INVESTIGATION IN MODELING A LOAD-SENSING PUMP USING DYNAMIC NEURAL UNIT BASED DYNAMIC NEURAL NETWORKS

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Mechanical Engineering

University of Saskatchewan

Saskatoon

By

Yuwei(Leslie) Li

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mechanical Engineering

University of Saskatchewan

Engineering Bulidings

57 Campus Drive

Saskatoon, Saskatchewan S7N 5A9

Canada

ABSTRACT

Because of the highly complex structure of the load-sensing pump, its compensators and controlling elements, simulation of load-sensing pump system pose many challenges to researchers. One way to overcome some of the difficulties with creating complex computer model is the use of “black box” approach to create an approximation of the system behaviour by analyzing input/output relationships. That means the details of the physical phenomena are not so much of concern in the “black box” approach. Neural network can be used to implement the black box concept for system identification and it is proven that the neural network have the ability to model very complex behaviour and there is a well defined set of neural and neural network structures. Previous studies have shown the problems and limitations in dynamic system modeling using static neuron based neural networks. Some new neuron structures, Dynamic Neural Units (DNUs), have been developed which open a new area to the research associated with the system modelling.

The overall objective of this research was to investigate the feasibility of using a dynamic neural unit (DNU) based dynamic neural network (DNN) in modeling a hydraulic component (specifically a load-sensing pump), and the model could be used in a simulation with any other required component model to aid in hydraulic system design. To be truly representative of the component, the neural network model must be valid for both the steady state and the transient response.

Due to three components (compensator, pump and control valve) in a load sensing pump system, there were three different pump model structures (the pump, compensator and valve model, the compensator and pump model, and the “pump only” model) from

the practical point of view, and they were analysed thoroughly in this study. In this study, the DNU based DNN was used to model a “pump only” model which was a portion of a complete load sensing pump. After the trained DNN was tested with a wide variety of system inputs and due to the steady state error illustrated by the trained DNN, compensation equation approach and DNN and SNN combination approach were then adopted to overcome the steady state deviation.

It was verified, through this work, that the DNU based DNN can capture the dynamics of a nonlinear system, and the DNN and SNN combination can eliminate the steady state error which was generated by the trained DNN.

The first major contribution of this research was in investigating the feasibility of using the DNN to model a nonlinear system and eliminating the “error accumulation” problem encountered in the previous work. The second major contribution is exploring the combination of DNN and SNN to make the neural network model valid for both steady state and the transient response.

ACKNOWLEDGMENTS

I would like to express my gratitude to Drs. Richard Burton and Greg Schoenau for their invaluable guidance, advice and encouragement throughout the research of the project. Additionally, a particular gratitude is also owed to Engineer, Doug Bitner, for his substantial assistance in the whole project.

Heartfelt thanks to my families: my father, my sister and parents in-law for your encouragement during this study. Special thanks are given to my husband, Yan Wang, for his everlasting love and support.

I acknowledge the financial assistance in the form of a scholarship from the University of Saskatchewan and support from NSERC research grants.

To my parents, Aiju and Yingcai

TABLE OF CONTENTS

page

LIST OF TABLES

Table page

LIST OF FIGURES

Figure page

LIST OF ABBREVIATIONS

DNU	The dynamic neural unit
DNN	The DNU based dynamic neural network
SNN	The static neural network
FFSNN	The feedback and feedforward static neural network

NOMENCLATURE

S_ω	Pump rotation speed	$rads/\sec$
Q_l	Pump leakage	$m^3/\sec$
P_s	Pump output pressure	P_a
Q_p	Pump flowrate	$m^3/\sec$
Q_s	Pump output flowrate	$m^3/\sec$
R_l	Leakage resistance coefficient of pump	$m^3/(P_a \cdot \sec)$
θ	Swash plate angle	$rads$
P_c	Pressure of control line	P_a
T_{lp}	Torque on yoke due to load pressure in the output line	$N \cdot m$
T_{spr}	Torque on yoke due to return spring	$N \cdot m$
T_{yk}	Torque on yoke due to control piston	$N \cdot m$
T_{yd}	Torque on yoke due to viscous damping and friction	$N \cdot m$
A_{cp}	Area of control piston face	m^2
k_{spr}	Angular spring constant of return spring	$N \cdot m/rads$
R_{yd}	Lumped damping resistance of swash plate assembly	$N \cdot \sec$
I_{yk}	Effective mass moment of inertia of yoke	$kg \cdot m^2$
$T_{spr}(0)$	Pretension of return spring	$N \cdot m$
k_{pri}	Coefficient of load torque	

k_{pr1}	Coefficient of load torque	$N \cdot m$
k_{pr2}	Coefficient of load torque	$N \cdot m / P_a$
k_{pr3}	Coefficient of load torque	$N \cdot m /(P_a \cdot rads)$
k_{pr4}	Coefficient of load torque	$N \cdot m \cdot \sec$

CHAPTER 1 INTRODUCTION AND OBJECTIVES

1.1 Introduction

Flow control in hydraulic systems is a very common method of varying the speed of an actuator (load). Essentially, there are three ways in which flow control can be accomplished: using a fixed displacement pump with varying shaft speed, a variable displacement pump or a pressure compensated orifice opening. In the first case, a variable displacement pump can be used to change the flow rate by varying the swash-plate angle of the pump or by changing the input shaft speed from the driving motor. This type of flow control is very efficient because energy losses across a controlling orifice are avoided. However, controlling the swash-plate angle or the shaft speed can be quite complex and in some instances, can introduce additional source of inefficiencies [Kim, et al., 1987 (1); Kavanagh, et al., 1990].

The second configuration is one of a pump and controlling orifice (via a valve downstream from the pump). If the pump pressure is restricted by a pressure regulator, then the pressure drop across the valve is a function of the load pressure and hence the pressure drop can vary significantly during operation. To obtain flow control, some measure of flow (or actuator velocity) is required. If the pressure drop across the orifice is controlled by forcing the pump pressure to follow the load pressure at a predefined and reasonable pressure difference (typically1.38 MPa), then flow control can be achieved and the energy losses minimized. This configuration still results in a pressure drop across the controlling orifice which translates to an inherent energy loss, albeit small. This pressure drop can also be used as a controlling signal within the valve or it

can be fed back to a control valve at the pump to improve transient responses and reduce the energy losses. “Load-sensing systems” are developed from this configuration.

One of the many advantages associated with these systems is the high energy- saving potential; that is, the pump attempts to match the power requirements to the changing load power requirements with minimal control losses (details are presented in Chapter 2). Because of the potential energy savings, load-sensing systems have found increasing usage in fluid power applications, especially in automobiles and off-road equipment [Book, et al., 1997]. Since the load-sensing system is a feedback nonlinear system, stability problems have been reported [Krus, 1988; Wu, et al., 2002; Lantto, et al., 1990; Kim, et al., 1988]. In addition, interactions between loads supplied by a common load sensing pump are sometimes encountered. To reduce instability and interaction problems, a common approach is to model the system, to investigate theoretically such problems and to provide means of reducing or eliminating these effects. This requires accurate models of the components.

The most important component in a load-sensing system is the pump. This component has received much attention from the research community as well as industry and has been the subject of many publications over the decades. This research has shown that because of the nonlinear behavior of pumps and the sensitivity of parameters to operating conditions [Kavanagh, et al, 1990; Zeiger, et al., 1986], derivation of a reliable and comprehensive pump model which is valid over a wide range of operation conditions is a challenge. It was this challenge that motivated the research presented in this thesis.

1.2 Models of Load-sensing Pumps

There are several methodologies which have been used for modeling hydraulic systems: modelling using mathematical physical relationships, modelling using system identification techniques or modelling using neural network approaches. Traditional mathematical modeling methods model the nonlinear behaviour of hydraulic dynamic systems and often involve mathematical modeling procedures based on the observation of physical relationships among the associated components [Ivantysyn, et al., 2000; Manring, N.D., 2005; Richards, et al., 1989]. System identification assumes a mathematical relationship between the input and output and uses identification techniques to establish the order and coefficients for the model [Soderstrom, and Stoica, 1989; Bellman, and Astrom, 1970]. Artificial Neural network (ANN) approaches assume a "black box" relationship between the input and output and simply capture/learn the input/output behavior for a wide range of operating conditions [Chen, et al, 1997; Narendra, et al, 1990;]. Normally, no attempt is made to associate the model with physical parameters or relationships.

In mathematical modeling approaches, the derivation of a suitable model for a practical component requires engineering knowledge of the system behaviour combined with mathematical properties of the model. In addition, all parameters, such as those describing physical properties of components and of the fluid medium must be supplied or obtained experimentally. For nonlinear systems, linearization techniques can be employed to simplify nonlinear modeling procedures; however, this places a constraint on the applicability of the model since the model obtained is only valid over a very limited region of operation. When this approach has been applied to load-sensing

systems, the model tend to be either overly simplified or extremely complex [Ivantysynova, 2003; Krus, 1988; Erkkila, 1999; Book, et al., 1997; Wu, et al., 2002].

System identification is an alternative to mathematical modeling. System identification is an experimental approach to the determination of the model of a real system based on the observed input-output relationships. Linear dynamic system identification has been reasonably well understood, and parameter estimation methodology has been systematically established [Astrom, et al., 1971], because simple linear forms of model structures have been assumed. In the field of nonlinear dynamic system identification, the choice of a suitable model structure, at present, still remains as a challenging topic in this area. In general, it is not practical to talk about the identification unless a specific model structure is imposed.

Although system identification approaches have not been a tool in the modeling of a load sensing pump to date, identification approaches have been applied to the modeling of other hydraulic systems. As an example, Habibi [2004] and Chinniah [2001] applied this approach to the modeling of an electrohydraulic actuation system. Their model represented relationship between the pump angular speed to the actuator displacement.

The third category of modeling is based on neural network technology. In recent years, many researchers have focused on applying neural network models for system identification/control and on improving performance of these neural network models. There are many published works in this area [McNamara, et al., 1997; Watton, et al., 1997; Bailey, et al., 2002; Oysal, 2005]. There have been several studies done on the use of ANN's or Multi-layer neural networks to model load-sensing pump [Hindman, 2002; Xu, et al., 1996; Xu, et al., 1997; Lamontagne, 2003]. The philosophy of these studies

has been that the neural network model is a “black-box” which has captured the actual input/output relationships for all operating conditions. Simulation studies by Xu examined the usefulness of various neural network structures for modeling non-linear dynamic systems and developed model structure error analysis. The feedback path in the recurrent model structures introduced dynamics into the model, but at the same time produced an “error accumulation problem” [Xu, et al., 1997]. Because the neural network model uses previous output as inputs, and error in the outputs affected all of the neurons at future time steps.

One way to overcome this problem was investigated by Lamontagne [2003]. His approach was to model the system using only delayed inputs. With input delays, the output of the network is never fed back to the input so errors in network output are not propagated to the next time step, and there is no accumulation of error. The disadvantages of this method are that the dynamic response is a less accurate reproduction of the actual output signal and that the modeling of dynamics is limited to behaviors that take place in time span shorter than that covered by the input delays.

A problem that was encountered by Xu and Lamontagne was the appropriate choice for the inputs and outputs to the networks. Xu examined two models, a single-input and single-output model (SISO) and a two inputs and one output model. For the SISO model, the load sensing valve was assumed to have a fixed opening which severely limited the flowrate range (and hence validity) of the SISO model. For Xu’s second configuration, the model was valid only for that particular controlling valve chosen. In the study by Lamontagne, two model structures were investigated: a combined pump/valve model and a stand-alone pump model (comprised of a compensator and

pump). As with Xu, the first model was valid only for that particular controlling valve chosen. For the stand-alone pump model, the ANN pump model performance was poor at low frequencies especially under steady-state conditions.

In summary, the literature indicates that the model of a load-sensing system remains an area which requires much further study. The ANN approach is appealing because it does not require extensive testing to determine parameter values or model order (which can change with varying operating conditions). The experience by Xu and Lamontagne clearly indicated the potential for using ANN's in this capacity but also established limitations and difficulties. For the modeling of a load-sensing pump, it is highly desirable to pursue a new methodology to develop a pump model using a neural network approach with a reasonable structure.

1.3 Objectives

As stated above, there are many challenges that exist in trying to use neural networks to model dynamic systems. It is necessary to examine newer structures of ANN's which integrate some dynamic properties within the basic neuron structure. Such a neuron is called a Dynamic Neural Unit (DNU) and has been extensively studied by Gupta [Gupta, et al., 1992 and 1993; Song, et al., 1999] for control purposes. Therefore, the objectives of this thesis are:

- To find a more applicable model structure (that is, what are suitable inputs and outputs) for a load-sensing pump which will allow the model to be independent of the operation of the load sensing orifice and controlling positions.

- To apply DNU methodology to the neural network based pump model and to determine the new structure's ability to eliminate the error accumulation problems experienced in previous research.
- To develop, train and test an ANN based load-sensing pump model which can represent both the steady state and transient response of the pump over an expected operating range.

1.4 Organization of This Thesis

In Chapter 2, a hydraulic load-sensing pump is introduced and the causal relationships between the components are discussed. The pump model structure investigated in this study is determined.

Chapter 3 reviews the previous work in system modeling using a static neural network approach. The dynamic neural units (DNUs) which were adopted in this study are also presented. The configuration of the dynamic neural network used for the load sensing pump is determined.

Chapter 4 presents the simulation results of the neural network model and identifies some problems in the results.

Chapter 5 explores two modification methods to compensate for the problems encountered with the DNN model. A series of validity tests for the modified model are examined.

Chapter 6 summarizes and concludes the study and forwards suggestions for the future work.

CHAPTER 2 LOAD-SENSING HYDRAULIC SYSTEM

2.1 Introduction

In this chapter, the operation of the load-sensing pump is described and the relationships between the components in a typical load-sensing system are discussed. In using a “black-box” modeling technique employing neural networks, it is very important to divide the system into reasonable and relevant component models. This determines how the models can be integrated into a complete system model.

2.2 Load-Sensing Pump System

A load-sensing pump is designed to maintain a fixed pressure drop across a controlling orifice and single load in order to control flow in an energy efficient manner [Moller, 1990]. Such a pumping system is often referred to as a pressure compensated load sensing flow control system. These types of pumps are commonly found in forestry and off - highway equipment. A schematic circuit of a load sensing pump and single load is illustrated in Figure 2.1.

The goal of the load-sensing system is to keep flow to the load constant for a given control valve setting independent of disturbances in P_L or P_s for acceptable pressure differences across the controlling orifice. This is accomplished by the load-sensing compensator (A in Figure 2.1), which adjusts pump displacement (pump flow) until the pump outlet pressure, P_S, is greater than the load pressure by a fixed amount. If the pressure drop ($P_S - P_L$) across the valve (B in Figure 2.1) is always held constant by

controlling the pump flow and hence P_S, then the flow rate through the valve, B, dependents solely on the load valve position determined by the valve input X_V [Lamontagne, D., 2001; Lantto, et al., 1993; Lantto, et al., 1990].

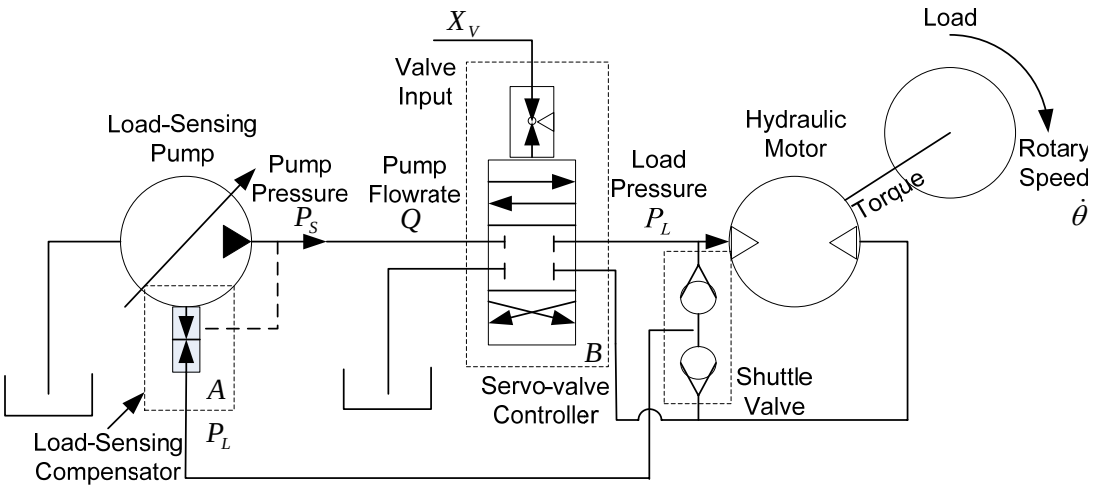


Figure 2.1 Schematic diagrams of a load sensing pump and single load.

Assume the system is at steady state conditions and the compensator is at the closed position. A force balance between hydraulic pressure (P_S and P_L) force and spring force exists. Consider Figure 2.2. Let P_L increase from the steady state condition. First, the pressure drop ($P_s - P_L$) across the load orifice decreases which results in a decrease in Q_L. This is evident by examining the orifice flow equation where $A(X_V)$ and C_d are assumed constant:

$$Q_L = C_d A(X_V)\sqrt{\frac{2(P_S - P_L)}{\rho}}. \qquad (2.1)$$

The increase in P_L is sensed at the compensator and creates a force imbalance across the compensator spool (See Figure 2.2). This causes the spool to be pushed to the right. The compensator spool orifice opens and ports pressurized fluid to tank.

Consequently, P_c, the control piston pressure, decreases, which results in a force imbalance (the difference between the spring force and the control piston "pressure force") on the swash plate. The swash plate angle, θ, increases producing an increase in pump flow to the load orifice. This, in turn, increases P_s which is also sensed by the compensator orifice. The force imbalance on the compensator spool is now reduced and a new equilibrium state is reached. The pressure drop across the load orifice is re-established and flow control is reestablished.

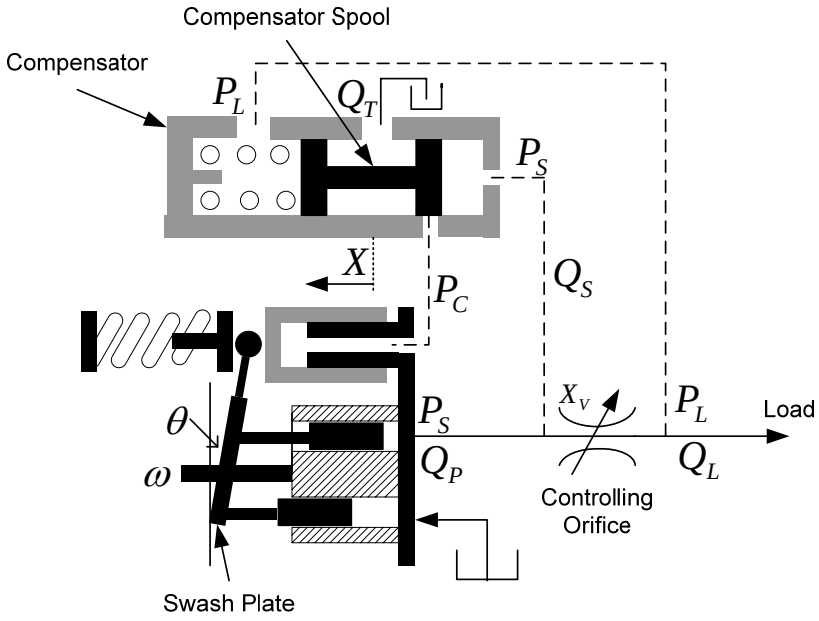


Figure 2.2 Schematic diagram of a load sensing pump.

Because of the complexity of operation, combined with many inherent nonlinear behavioral characteristics, accurate modeling and simulation of load sensing pumps have been a major challenge for researchers and designers. Krus has approximated the performance of load sensing systems in a dynamic sense using a very simple linearized model which, although very useful for preliminary studies, is not very accurate for precise or interaction type studies [Lantto, et al., 1993 and 1990]. In the study of Wu

[2005], the performance and stability of a load-sensing system was investigated over three different operating regions. To accomplish this, the nonlinear characteristics were linearized about various operating points. He established that the performance and stability were very dependant on these operating conditions. His model was very complicated but its validity was established experimentally over certain frequency ranges.

In contrast to the traditional mathematical modeling approaches of hydraulic systems, several researchers from the University of Saskatchewan have investigated the feasibility of using a neural network approach for modeling load sensing system components. Xu [1997 (1)] and Lamontagne [2003] used various neural network (NN) morphologies to capture the static and dynamic performance of a load sensing pump. They showed that the applicability of their NN based load-sensing pump models was limited by the morphologies of the ANN adopted in their simulation as well as the model structure that was assumed for physical system. It was clear from Xu and Lamontagne's studies that modeling of the load-sensing pump using NN structures needed to be further investigated.

This model structure problem is now considered with the NN morphology issue being addressed in the next chapter.

2.3 Model Structure and Justification

The load-sensing pump system is comprised of four main components: the load-sensing pump, the pump compensator, the control valve and the load. In a given system, there are additional minor components such as lines, valves, and filters etc. that affect the performance of the system. It is desirable to separate components as much as

possible when modeling to allow the user to construct system models simulations from the various components. This avoids having to model a large number of possible combinations for a single component; each component is therefore modeled with a unique set of inputs and outputs. These models are then linked together in much the same fashion as a physical hydraulic system would be in construction. Separating the component models into their basic form also allows for much more flexibility in overall system modeling changes. For example, consider a standard model of a load-sensing system (Figure 2.2). If the pump and valve were modeled together as a SISO structure, then any change in line length which connects them would require an entirely new NN model to be trained.

Consider the four main components (pump, compensator, control valve and load) shown in block diagram form in Figure 2.3. If the model is based on NN approaches in which only inputs and outputs of the component model are considered, there are several combinations in which the system could be modeled. These are now considered.

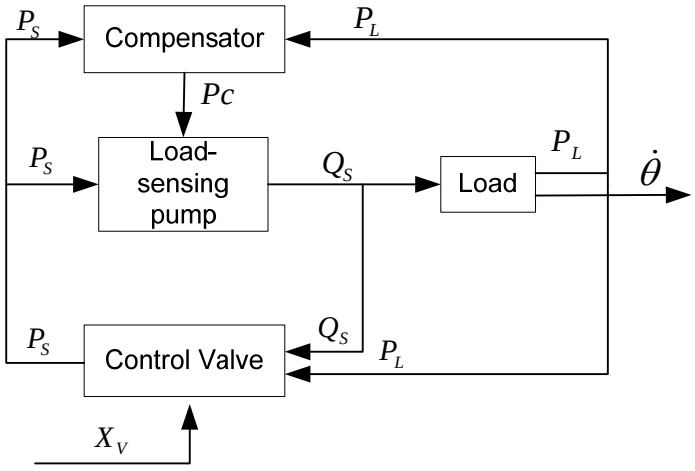


Figure 2.3 Causal block diagram of a load sensing pump.

2.3.1 Pump & Compensator & Valve Model

The most straightforward way to model the system would be to consider X_v and P_L as inputs and $\dot{\theta}$ as the output in a single neural network model. The pump, compensator and valve would all be grouped together into a single "black-box" model and only the inputs and outputs external to these components would be modeled (as shown in Figure 2.4 (a)). The interactions between the compensator, pump, and valve would all be internal to the model and the causal diagram can be reduced to a single block as shown in Figure 2.4(b). The model thus represents a single component with multiple "hidden" subcomponents having two inputs and a single output.

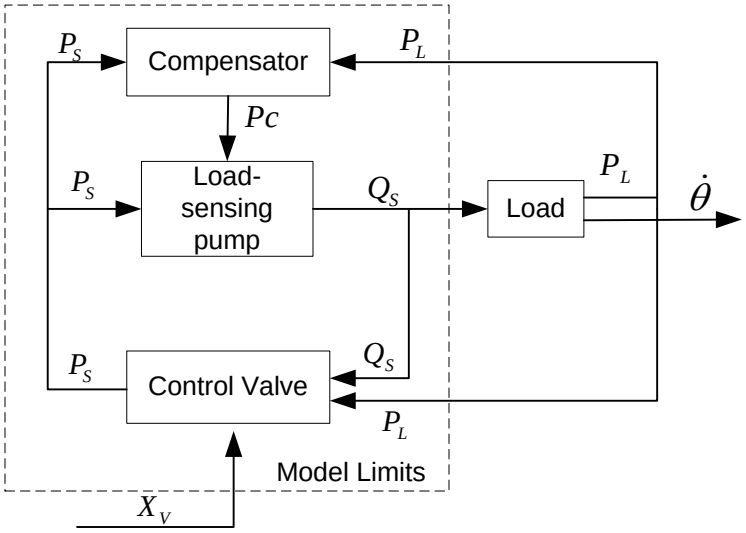


Figure 2.4 (a) Causal block diagram of grouped system components.

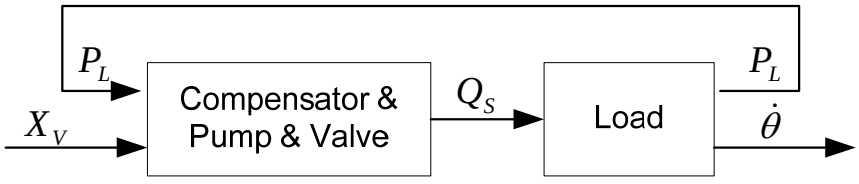


Figure 2.4 (b) Block diagram of pump/compensator/valve sub-system.

Ddifferent loading conditions would have to be created to train the NN based pump and control valve combination. Theoretically, loads could be represented by a set of data points allowing a spectrum of loading conditions to be simulated. From a practical point of view, the problem with this grouping is that the data is only valid for the particular control valve chosen. Any change in the valve requires creating an entirely new trained model, and as such, this model would have little application for design, for example. In addition, an experimental system must be set up for each loading condition (inertial, viscous, gravity etc) that the load sensing system would be expected to encounter. Simulating experimentally these conditions could be a challenge in itself.

In the studies by Lamontagne [2001] and Xu [1997], this model structure was indeed investigated experimentally and was only valid for the particular control valve chosen. In the study by Xu [1997], the valve setting was fixed in the experimental investigation, reducing the model to a single-input and single-output (SISO) form. As a consequence, the fixed opening setting of the control valve limited the range of flow rate operation the SISO model.

2.3.2 Pump & Compensator Model

Another model structure which separates the load and the control valve from the pump model is shown in Figure 2.5 (a) and in block diagram form in Figure 2.5 (b).

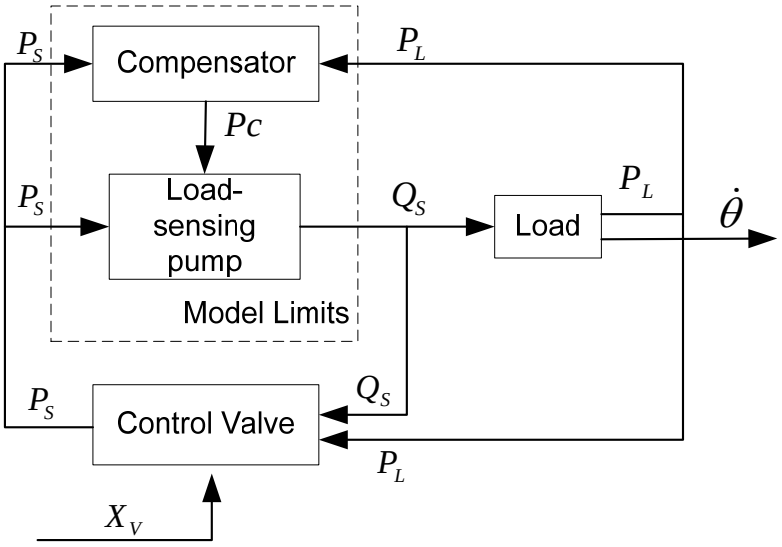


Figure 2.5(a) Causal block diagram of grouped system components.

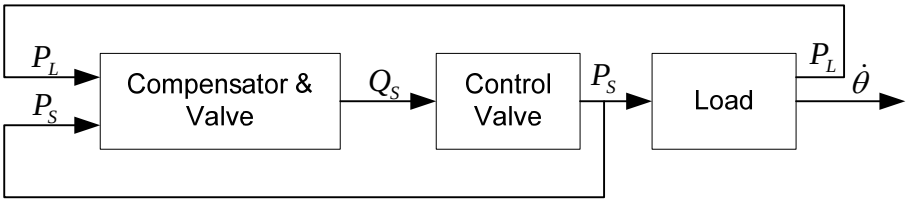


Figure2.5 (b) Block diagram of pump/compensator sub-System.

In this model, P_L and P_s are considered as input, and the flowrate, Q_s is the output. The control valve and load are isolated form the model. The causal diagram of the compensator and the pump can be reduced to a single block as shown in Figure 2.5(b). This would allow different combinations of pump, valve and load models to be combined into different systems for simulation.

A problem arises from this model structure. The compensator in the load-sensing pump adjusts pump displacement until $P_S - P_L$ balances the compensator spring force. That means if $P_S - P_L$ across the pressure compensator meets the required value set by the spring, then the compensator spool is at the closed position. If the desired flow rate is

changed via the control valve, and if equilibrium conditions for $P_S - P_L$ are reestablished, the compensator spool is once again at the closed position. Hence there is no unique relationship between inputs P_L and P_s and the output Q_S. That is, the relationship between the inputs and output are not unique. The neural network approach can not be used to model this situation.

An additional consideration is posed by an experimental limitation. If the pump and compensator model is to be trained using experimental data with the control valve present, there are many practical problems generating the training data. Since P_L is an input, some laboratory technique must be established to simulate various loading conditions. This can be done by using a variable orifice or a pressure control device downstream of the control valve. In reality, this means the inputs P_S and P_L cannot be made independent; a requirement for NN input training. Thus the training of the model must be done such that P_S and P_L are varied independently. This can be done hydraulically using various flow or pressure servo-valves but as Xu and Lamontagne observed, the frequency response of these valves have to have cutoff frequencies significantly higher than the actual load sensing components they are sensing. This did pose a challenge given the limitation of the valves available in the laboratory.

An interesting problem arises when P_S and P_L are, in fact, made independent. If the difference between P_S and P_L is larger than the set point on the compensator, the compensator valve is fully opened at one of its extremities. Thus the swash plate of the pump is either fully stroked (maximum flow rate) or is at zero stroke (zero flow rate). Thus, having too large of a difference between P_S and P_L beyond the compensator

spring setting will result in saturation conditions for the physical pump which do not reflect realistic behavior during normal load sensing control conditions. This was a point that both Xu and Lamontagne had to consider in their experimental testing.

In the study by Lamontagne [2001], the test results of the compensator pump model indicated that the model did performed well at high frequencies of 3Hz, but the pump model performance rapidly deteriorated at low frequencies, especially under steady-state conditions.

2.3.3 "Pump Only" Model

Based on the observation and conclusions drawn by Xu and Lamontagne, it was decided that the third structure, defined as a pump only model, would be investigated in this study. This structure is as shown in Figure 2.6. In this model, the pump is isolated form the load-sensing compensator, the valve and the load. This would allow the model to be combined with different control valve, load and compensator models (obtained in any form, including NN approaches).

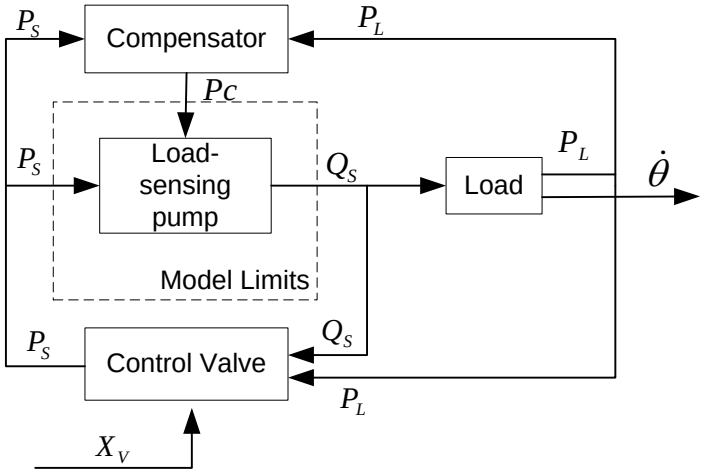


Figure 2.6 Causal block diagram of "pump only" sub-system.

As the pump model is separated from the compensator, the model inputs would be P_S and P_C, and the output Q_S. As before there are two inputs and one input. What is important in this configuration is the fact that the relationship between the inputs and output is unique, since the pressure P_C controls the swash plate directly and P_S dictates the back torque on the swash plate and the pump leakage; further, the relationship between P_S and P_C is also independent. This input parameter selection eliminates the "unique relationship" problem in the pump and compensator model experienced by Xu and Lamontagne.

From an experimental implementation point of view, the generation of P_S and P_C would be subjected to the same frequency response constraints defined by Xu and Lamontagne. However, since P_S and P_C require very little flow in their lines (see Figure 2.4), pressure servo-valves could be used which tend to have excellent frequency response characteristics.

As a final note, it should be mentioned that the approach using P_S and P_C as inputs is similar to that taken by others in dynamic pump modeling [McNamara, et al., 1997].

2.4 Summary

In this chapter, the causal relationships between the components in a typical load-sensing system were discussed. For the various pump, compensator and valve structures, the models were shown to be only valid for the particular controlling valve chosen. For the pump and compensator model problems associated with "unique relationship", "dependent inputs" and "experimental data collecting" were experienced by Xu and

Lamontagne. Compared to these two structures, the “pump only” model avoided many of the issues discussed and hence, in this study, was the only structure considered.

CHAPTER 3 DYNAMIC NEURAL NETWORK APPROACH

3.1 Introduction

Artificial neural networks (ANNs) have been the subject of significant research from various investigators in the nonlinear system identification and control area. An artificial neural network is a massive “net” that consists of a number of identical computing units referred to as neurons or nodes that are connected together. Many variations on neural network morphology or structure have been used for system identification and control applications [Chen, S. et al, 1990; Xue, et al., 1995; Azam, et al., 1997]. These morphologies depend on the way the neurons are interconnected and the operations in each neuron.

A classic “static” neuron can be considered as a processing element that sums the weighted inputs and produces an output which corresponds to a predefined value for that particular input. The transformation from weighted inputs to output is accomplished by a particular function called an “activation function” or “activation operator”, which can either be linear or nonlinear in form. This static neural model has no feedback connections and as such, this model has no memory of dynamics. The neural output is solely determined by the current inputs and values of the synaptic weights (the “gains” that connect the outputs of a neuron to the inputs of the next). Neural network structures based on this model describe the synaptic connection by a single weight parameter vector [Rao, 1994]. However, biological neural systems are considered to be composed of structures with dynamic connections which are manifested in the temporal properties of the synapse along with such processes as impulse transmission and membrane excitation [Fukushima, et al., 1983; Anderson, 1983; Hopfield,1984]. In order to emulate some of these dynamic functions, such as learning, adaptation, memory and recall, and to reflect the

dynamics of the biological neuron in a better way, a new neuron architecture, defined as a dynamic neural unit (DNU) was proposed by Gupta and Rao [1992 and 1993]. The DNU consists of internal feedforward and feedback weights and a nonlinear activation function. The built-in internal feedforward and feedback weights are what distinguish from the conventional structure of an artificial neuron from the DNU.

A significant property of an ANN is that it has the capability to “learn”, or to be trained, to recognize predefined relationships between inputs and outputs. During the learning (training) process, the ANN is subjected to a series of input output data pairs; the weights in the ANN are adjusted via a “training” algorithm until the difference between the ANN output and the target (predefined) output reaches an acceptable value. After the training stops, the weights are fixed at the values which “minimized” the training error during the training process. The ANN is then evaluated by subjectecen it to new inputs which were not used in the training process. The trained ANN output and the real system output for these new inputs are compared to see if the trained ANN accurately represents the systems. This is defined as the “test” process. This test process is very important in establishing the ability of the trained ANN model to physically represent the plant.

It should be pointed out that for a given neural model set, the weights do not reflect any physical considerations in the unknown system, and hence are viewed simply as a means for adjusting the model to fit the input/output data observations.

There are two different styles of training: instantaneous training and batch training [Haykin, s., 1994]. In instantaneous training, the weights of the ANN are updated each time an input is presented to the network. In the batch training, the weights are only updated after all of the inputs over a defined number of time steps, have been presented to the network. The most

common way to train the networks is using backpropagation techniques (based on gradient descent approaches) [Widrow, et al., 1990]. One such approach will be examined in this section.

In this chapter, the structure of the classic static neuron and the use of static neural networks in modeling hydraulic components are reviewed. The new neuron architecture, the dynamic neuron unit (DNU), is introduced. Several dynamic neural unit morphologies are proposed, and the neural structure which is suitable for the modeling of the "pump only" model is selected.

3.2 Static Neuron

A simple model of a static neuron is illustrated in Figure 3.1 [Rosenblatt, 1959; Widrow, 1960]. This structure of an artificial (computational) neuron receives inputs either from other neurons or from sensors. A weighted sum of these inputs constitutes the argument of a "fixed" activation function, $\psi[\bullet]$ (either linear function or nonlinear function) as shown in Figure 3.1. The resulting output value of the activation function is the neural output.

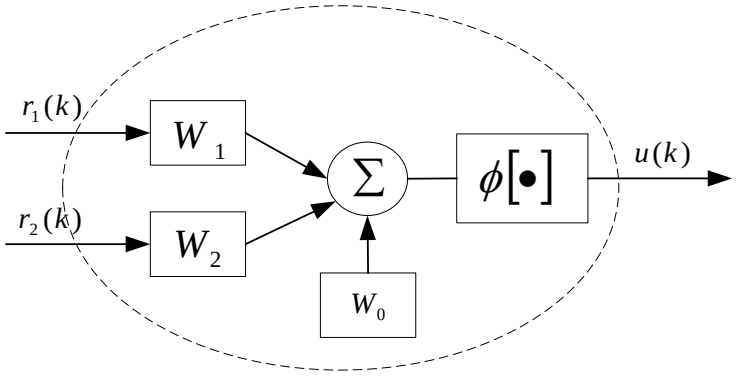


Figure 3.1 Structure of an artificial neuron.

The static neural model shown in Figure 3.1 can be integrated into a network configuration to form a number of neural networks structures. A static neural network, in general, consists of a number of neural layers (stages) where the output of one neuron forms an input to other neurons in the next layer. A static neural network is shown in Figure 3.2. In this figure, each shaded circle

represents the static neuron shown in Figure 3.1. Usually, these kind of structures are referred to as feedforward neural networks or multi layered neural networks (MNN)[Simpson, et., 1990; Hunt, et al., 1992; Hush, et al., 1993; Anderson, et., 1983; Wasserman, et., 1989; Hecht-Nielsen, et., 1988]. These feedforward neural networks respond instantaneously to inputs because they posses no dynamic elements in their structure. Therefore, feedforward neural network structures are often called static neural networks.

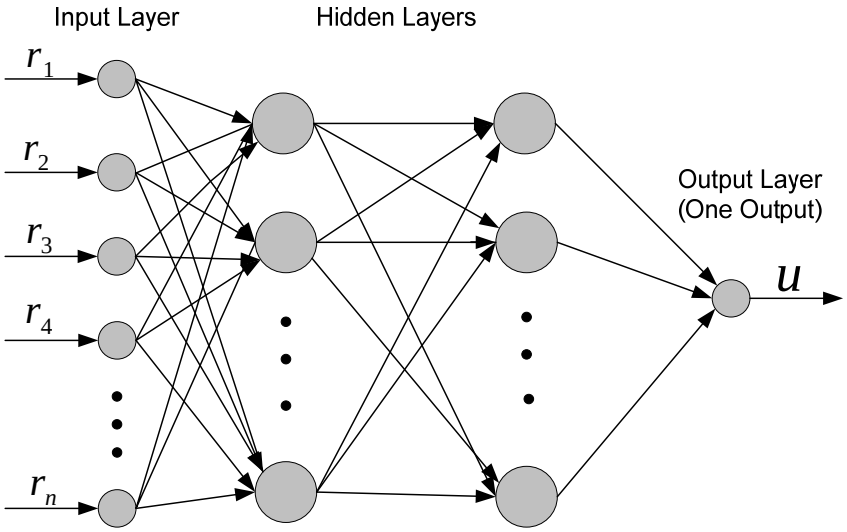


Figure 3.2 Structure of feedforward static neural network.

As an extension of static neural networks, dynamic neural networks using static neurons with external feedback and time delay inputs have been proposed [Hopfield, 1984; Narendra, et al., 1990]. A general topology of a dynamic neural structure is shown in Figure 3.3. This feedback based neural network consists of static neurons as the basic functional unit. The network output at any time step is dependent on previous outputs of the network; hence the recurrent structure allows the neural network model to represent system dynamic behaviour including higher-order characteristics. However, the neural network model uses previous outputs as inputs, and in system modeling approaches, any error in the outputs will affect the results at future time steps. This process has been defined as “error accumulation” by Xu [1997]. Because of the discrete

nature of the time delayed feedback, the smaller the implementation time step and the more time delay paths used, the larger the rate of error accumulation. This is evidenced by the studies of both Xu [1997] and Lamontagne [2001].

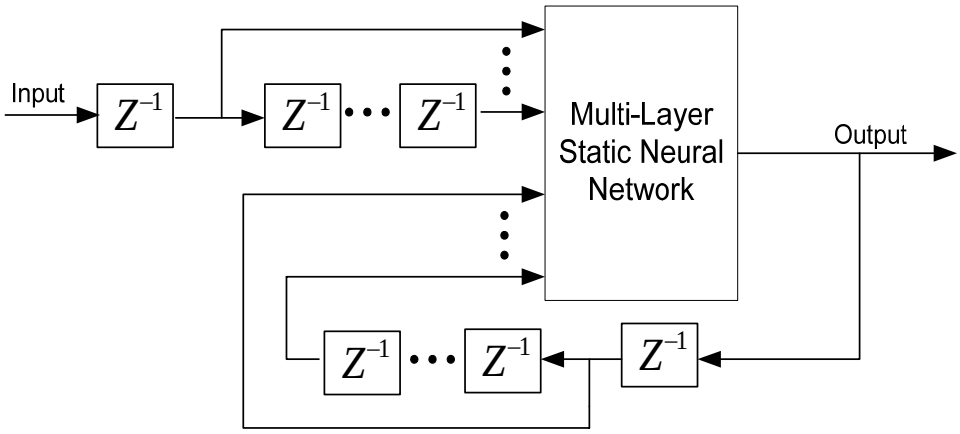


Figure 3.3 Dynamic neural network using static neurons.

To overcome the “error accumulation” situation, Lamontagne modeled the system using only delayed inputs (a feedforward-delayed input network) structure. With input delays, the output of the network is never fed back to the input, so errors in network output are not propagated to the next time step, and there is no accumulation of error. The results demonstrated that the input delay model had similar accuracy to the recurrent network under steady state conditions, but the dynamic response was less accurate. This result was expected as an input delay model is not able to model as wide a range of dynamic conditions as an output delay model [Lamontagne, 2001].

3.3 Dynamic Neuron Morphology

3.3.1Dynamic Neural Units

The studies by Xu and Lamontagne indicate that there are some morphology limitations in modeling a load sensing pump system using static neural networks. There have been many studies published in the literature on alternate dynamic neuron configurations. Some very

substantive research using a dynamic neural unit was conducted by Gupta, Deshpande, and Song for example [Gupta, et al., 1992, 1993 and 1995; Deshpande, et al 1998; Song, et al., 1999; etc]. The model in their studies was defined as the dynamic neural unit (DNU) which consisted of internal feedforward and feedback weights and a nonlinear activation function. The introduction of feedback introduces dynamic memory characteristics in the network and this results in faster convergence of the solution and better system characteristics.

One of the advantages of the dynamic neuron is that it reduces the dimension of the network and the amount of computational time required when it is used to model or control a dynamic system. In previous studies [Gupta, et al., 1993 and 1992; Deshpande, et al., 1998; Song, et al., 1999; Song, 2001; Srivastava, et al., 1998], the effectiveness of one DNU as an inverse dynamic controller was investigated. The results showed that a single DNU can be used to control linear and simple nonlinear systems whereas this was almost impossible for one static neuron.

Another advantage of the dynamic neuron is that it has the potential to overcome the error accumulation problem associated with system modeling using a feedback and feedforward static neural network (FFSNN). The number of FFSNN inputs is determined not only by the number of physical system inputs, but also by the number of feedback and feedforward delay paths (shown in Figure 3.4).The output with error is fed back to the FFSNN as a “derivative” input and results in an increase of error in the output which is the new “derivative” input in the next time step. This would eventually drive the FFSNN far away from the target output as the simulation time increases. The “error accumulation” problem, therefore, can not be eliminated by the static neural network alone.

With the DNU/DNN, the output feedback, the time delay inputs, and the system inputs are all summed and fed to the input of the nonlinear activation function. The DNU/DNN in this

configuration makes the DNU/DNN, in this particular case; behave similar to second order dynamic system. Like all other second order systems, the DNU/DNN keeps the output error within a certain range which is determined by the DNU/DNN structure and the parameters (weights) rather than the inputs and outputs. As a result, the output feedback will not affect the DNU performance as long as the training is stopped. The built-in internal feedforward and feedback time delays in the DNU make a single DNU a dynamic unit. This enables the DNU/DNN to capture the dynamics of the system without having to add the external feedback connection, as needed in the FFSNN, and also, makes the DNU/DNN superior to the static neural network in system modeling.

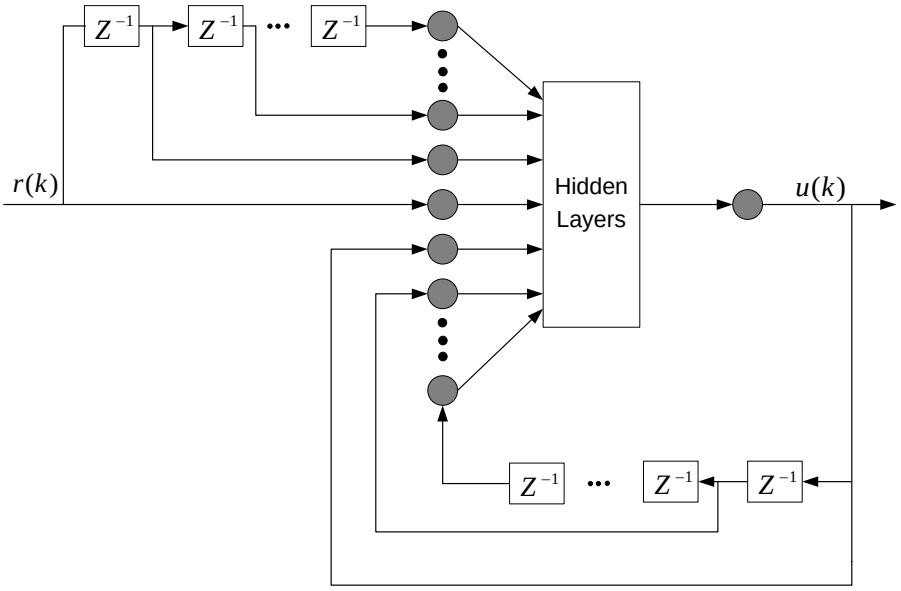


Figure 3.4 Structure of dynamic neural network using static neurons

Some of the earliest research on the development of the models of dynamic neurons was done by [Gupta, et al., 1993 and 1992; Deshpande, et al., 1998; Song, et al., 1999; Song, et., 2001; Srivastava, et al., 1998]. In all these studies, the dynamic neural units (DNUs) and the dynamic neural networks based on dynamic neural units (DNNs) were shown to be able to represent dynamic nonlinear systems. In most cases, DNUs/DNNs were used for adaptive control purposes

where the plant and the DNUs/DNNs were not separated during the control process; further, both the controlling of the plant and the training of the DNUs/DNNs were performed simultaneously. In some other cases, the use of DNUs/DNNs for "function approximation" was investigated, but after the "training" stopped, the trained DNUs/DNNs were not isolated from the desired function and were not tested. What this meant was that the networks were continuously responding (essentially changing weights) to the error signals from the last time step and were not really being trained to capture the dynamics of the system since the error signal was always present. This is an important difference to what is being attempted in this study in which the error signal is not present when the model is being used. The studies reported in the literature were "on-line" control and "on-line" identification where the error was available for continuous training. It is not clear from these studies how well the neural networks represented a physical system once the training stopped. The capability of DNUs/DNNs to model a nonlinear dynamic system needs to be further investigated and this is one of the objectives of the present study.

Some of the DNU model structures presented and studies in the literature are shown in Figures 3.5, 3.6, 3.7, 3.8 and 3.9. Following each figure, the input-state, output-state representations are also provided for completeness.

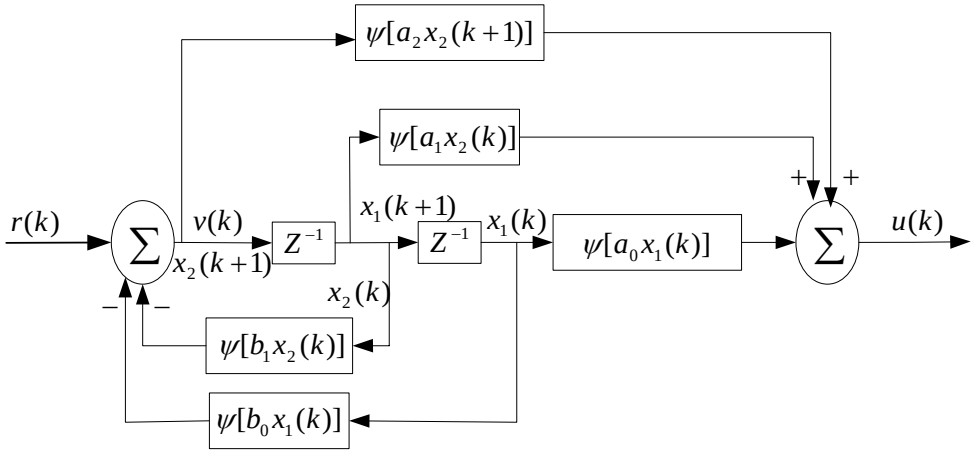


Figure 3.5 Structure of DNU-1 [Deshpande 1997].

The input-state-output representation of DNU-1 is expressed mathematically as:

$$x_1(k+1) = x_2(k), \tag{3.1}$$

$$v(k) = r(k) - \psi[b_1 x_1(k)] - \psi[b_2 x_2(k)], \tag{3.2}$$

$$x_2(k+1) = v(k), \tag{3.3}$$

$$u(k) = \psi[a_1 x_1(k)] + \psi[a_2 x_2(k)] + \psi[a_3 v(k)]. \tag{3.4}$$

In structure of DNU-1, the feedback and feedforward functional relationships are nonlinear.

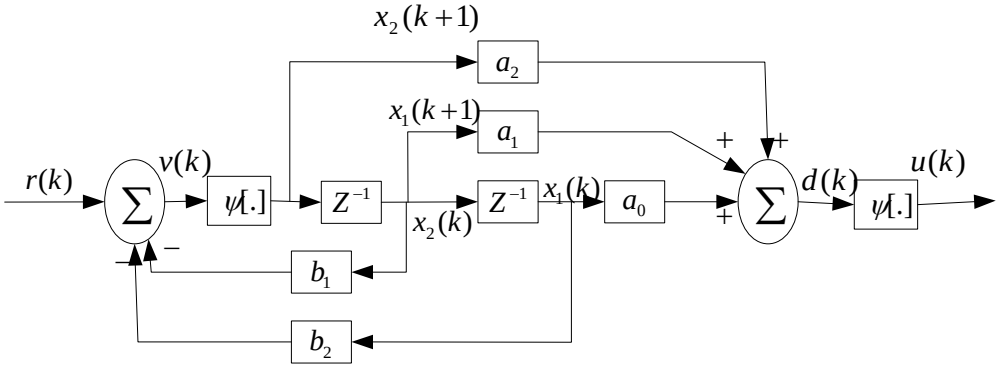


Figure 3.6 Structure of DNU-2 [Deshpande 1997].

The input-state-output representation of DNU-2 is expressed mathematically as:

$$x_1(k+1) = x_2(k), \tag{3.5}$$

$$v(k) = r(k) - b_1 x_2(k) - b_2 x_1(k), \tag{3.6}$$

$$x_2(k+1) = \psi[v(k)], \tag{3.7}$$

$$d(k) = a_0 x_1(k) + a_1 x_2(k) + a_2 x_2(k+1), \tag{3.8}$$

$$u(k) = \psi[d(k)]. \tag{3.9}$$

In the structure of DNU-2, the feedback and feedforward contain constant coefficients compared to DNU-1.

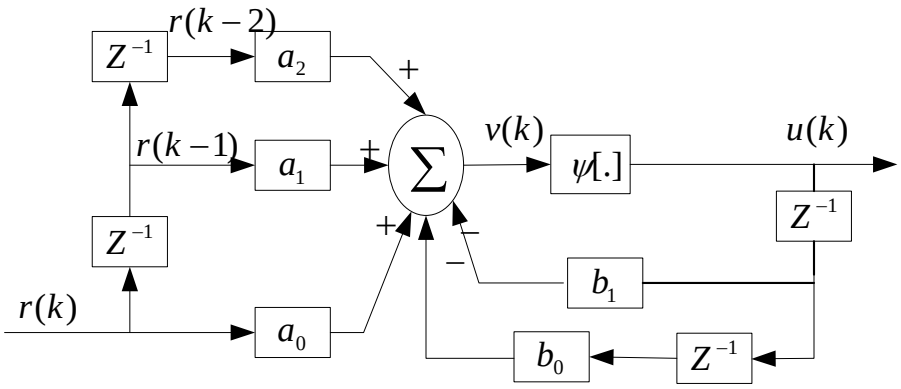


Figure 3.7 Structure of DNU-3 [Srivastava 1998].

The mathematical expressions are illustrated as follows:

$$v(k) = a_0 r(k) + a_1 r(k-1) + a_2 r(k-2) - b_1 u(k-1) - b_0 u(k-2), \tag{3.10}$$

$$u(k) = \psi[v(k)]. \tag{3.11}$$

In structure of DNU-3, the feedback and feedforward structure is different than DNU-1 and DNU-2. In addition, the feedback and feedforward coefficients are constants.

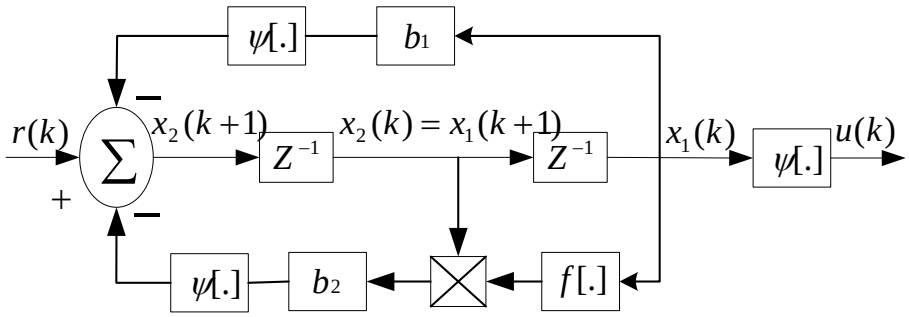


Figure 3.8 Structure of DNU-4 [Song 1999].

The input-output state representation of the DNU-4 is expressed as:

$$x_1(k+1) = x_2(k), \tag{3.12}$$

$$x_2(k+1) = r(k) - \psi[b_1 x_1(k)] - \psi[b_2 x_2(k) f(x_1(k))], \tag{3.13}$$

$$u(k) = \psi[x_1(k)], \tag{3.14}$$

where, $f(x_1(k))$ is a nonlinear function and is chosen as $f(x_1(k)) = 1 - x_1^2$. The basic structure of DNU-4 is significantly differently than DNU-1, DNU-2 and DNU-3. Some coefficients are constant and the feedback functional relationships are nonlinear.

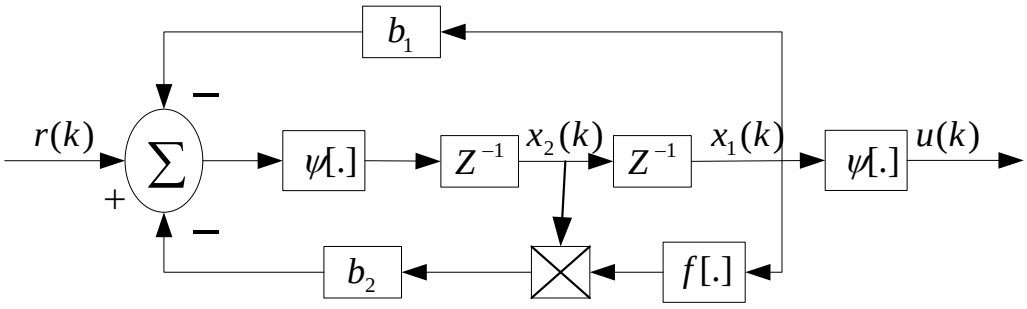


Figure 3.9 Structure of DNU-5 [Song 1999].

The input-output state representation of the DNU-5 is expressed as:

$$x_1(k+1) = x_2(k), \tag{3.15}$$

$$x_2(k+1) = \psi\left[r(k) - b_1 x_1(k) - b_2 x_2(k) f(x_1(k))\right], \tag{3.16}$$

$$u(k) = \psi[x_1(k)]. \tag{3.17}$$

where, $f(x_1(k))$ is a nonlinear function and is chosen as $f(x_1(k)) = 1 - x_1^2$. The structure of DNU-5 is similar to DNU-4 with the exceptions that the feedback contains constant coefficients.

Compared with DNU-1, DNU-2, and DNU-4, and DNU-5, DNU-3 has the following advantages:

(1) Due to the decrease of the total sigmoidal-type nonlinearities (only one), the computational requirements of DNU-3 are reduced considerably compared to those of other structures. If dynamic neurons are combined to form different kinds of dynamic neural networks to identify or control complex dynamic nonlinear systems, the computational requirement is very important.

(2) The feedback connections of the DNU-3 are taken after the nonlinear operation. Since the output of the nonlinear operation is bounded, this will increase the stability of the DNU-3.

(3) The two time delays (in both input and feedback path) in DNU-3 indicates that the DNU-3 is a second order dynamic unit, which, in this particular study, matches the order of the "pump only" model (in which the order of the "pump only" model was dominated by second order terms. Details are presented in Chapter 4).

Therefore, DNU-3 was adopted in this project.

Since the activation function plays a vital role in the application of the DNN, consideration should be given to the choice of the activation operator selection. Firstly, because the DNUs/DNN is used to approximate a nonlinear function, a nonlinear activation function should be adopted in which a nonlinear mapping from input to output is accomplished. Secondly, in the real pump situation, flow saturation arises when the swash plate of the pump is either fully "stroked" or is at zero- stroke; thus the nonlinear function needs an upper and lower threshold to meet the pump flowrate limitation requirement. Thus, a hyperbolic tangent function (shown in Figure 3.10) was deemed to be one of the most appropriate for this study.

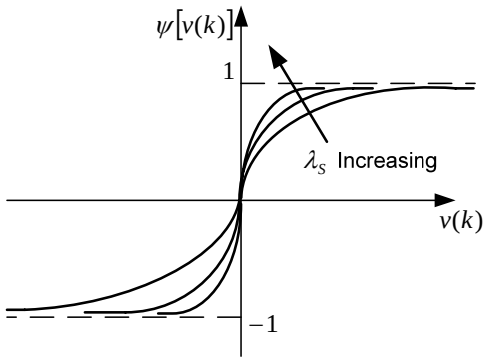


Figure 3.10 Hyperbolic tangent function

Mathematically, the activation function equation is given as

$$\psi[v(k)] = \left[\frac{e^{(\lambda_S v)} - e^{-(\lambda_S v)}}{e^{(\lambda_S v)} + e^{-(\lambda_S v)}} \right] = \tanh[\lambda_S v], \tag{3.18}$$

where λ_S is the gain and controls the slope of the activation function. If λ_S is too large, it may lead to instability. Inaddition, the error can increase. If λ_S is too small, the learning occurs very slowly. In the following section λ_S is fixed at 2. $v(k)$ is the function input.

3.3.2 Dynamic Learning Algorithm

Through computer simulations, the ability of a dynamic neural structure with the DNU as the basic computing node to represent nonlinear dynamic systems can be investigated. In this study, the dynamic system was the "pump only" model. However, before this ability can be established, it is necessary to discuss how the neural network is "trained". The training process is made more complex by the presence of time delayed feedback paths within the DNU.

The basic training process employed for this task is shown in Figure 3.11.

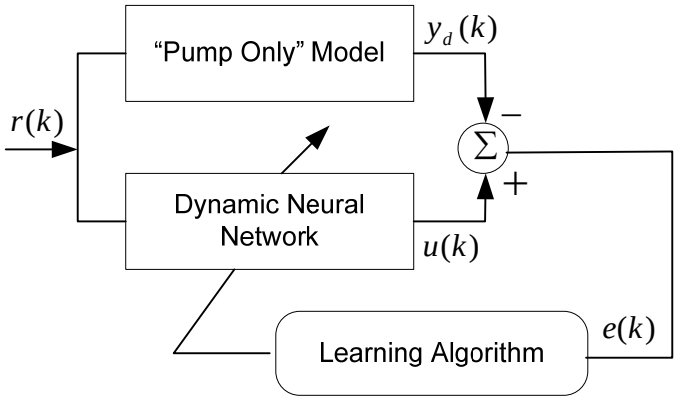


Figure 3.11 Scheme of learning for system simulation using dynamic neural network.

As shown in this Figure, the input signals are given to both the "pump only" model and the DNN, and the difference between their outputs is used to update the parameters (weights) in the DNN via the learning algorithm until the error is minimized or reaches a tolerance value. The least mean-square algorithm, based on the least mean-square error between the desired and actual

output, and the dynamic gradient descent algorithm described in this chapter, are adopted as the learning algorithm. These are now considered.

As the DNU-3 (shown in Figure 3.7) was selected to be the computing node in the DNN, the learning algorithm based on the DNU-3 is as follows.

Consider Figure 3.11. The error signal $e(k)$ is the difference between the desired output $y_d(k)$ and the actual output $u(k)$ at kth instants. It is defined as

$$e(k) = y_d(k) - u(k) . \tag{3.19}$$

The error signal drives the learning algorithm to update the feedforward weights a_i, i=0, 1, 2, and the feedback weights b_j, j=1, 2, in DNU-3 (Figure 3.7), such that the performance measurement function $E(\cdot)$ error is minimized. $E(\cdot)$ is given by:

$$E(k) = \frac{1}{2}\frac{1}{N}[\sum_{m=(k-N+1)}^{k} e^2(m)] \ , \tag{3.20}$$

where m is the current time step, N is the number that indicates the amount of past information used in the calculation of $E(k)$. If $N = 1$, then only the information at that step is used to calculate the error for the minimization process. This is called instantaneous training. If $N = 10$, then 10 steps are made, and the error summed for those ten steps, before minimization occurs. This is called batch training. Depending on the dynamic system being simulated, N could be set to one or a larger number following some initial learning with more frequent weight updates. In this study, since instantaneous training was used, N was equal to 1.

The learning algorithm is based on the dynamic backpropagation method [Narendra, et al., 1990; Deshpande, et al., 1996]. The adaptation equations for the adjustable parameters are described as follows:

$$a_i(k+1) = a_i(k) + \Delta a_i(k), \qquad i = 0, 1, 2, \tag{3.21}$$

$$b_j(k+1) = b_j(k) + \Delta b_j(k), \qquad j = 1, 2. \tag{3.22}$$

Using the gradient descent approach [Werbos, P.J., 1974; Gupta, et al., 1992], the adjustments in the feedforward parameters, $\Delta a_i(k)$ and in the feedback parameters, $\Delta b_j(k)$, are based on the following equations:

$$\Delta a_i(k) = -\eta_{ai} \frac{\partial E(k)}{\partial a_i(k)}, \qquad i = 0, 1, 2, \tag{3.23}$$

$$\Delta b_j(k) = -\eta_{bj} \frac{\partial E(k)}{\partial b_j(k)}, \qquad j = 1, 2, \tag{3.24}$$

where η_{a_i}, $i = 0, 1, 2$, and η_{b_j}, $j = 1, 2$, are the individual gains (learning rate) of the adaptable parameters of the neuron models, which determines the stability and speed of the convergence to the optimum values. In this study, all the values of learning rate were set to equal to 0.005.

Substituting Equation (3.20) (with $N = 1$) into Equation (3.23), the adjustments in feedforward weights can be written as

$$\Delta a_i(k) = -\eta_{ai} \frac{\partial E(k)}{\partial a_i(k)} = -\eta_{ai} \frac{\partial \{\frac{1}{2}[y(k) - u(k)]^2\}}{\partial a_i(k)} = \eta_{ai} e(k) [\frac{\partial u(k)}{\partial a_i(k)}]. \tag{3.25}$$

In the structure of DNU-3 (shown in Figure 3.7), $v(k)$, the input of the nonlinear function and $u(k)$, the DNU-3 output are described by the following equations

$$v(k) = a_0(k)r(k) + a_1(k)r(k-1) + a_2(k)r(k-2) - b_1 u(k-1) - b_2(k)u(k-2), \tag{3.26}$$

$$u(k) = \psi[\lambda_s \cdot v(k)]. \tag{3.27}$$

Substituting Equation (3.27) into Equation (3.25) yields

$$\Delta a_i(k) = \eta_{ai}(k)e(k)[\frac{\partial \psi[\lambda_s v(k)]}{\partial a_i(k)}]$$

$$= \eta_{ai}(k)e(k)[\frac{\partial \psi[\lambda_s v(k)]}{\partial v(k)} \frac{\partial v(k)}{\partial a_i(k)}]$$

$$= \eta_{ai}(k)e(k)[\lambda_s \frac{4}{(e^{\lambda_s v} + e^{-\lambda_s v})^2} \frac{\partial v(k)}{\partial a_i(k)}]$$

$$= \eta_{ai}(k)e(k)[\lambda_s (1-u^2(k)) \frac{\partial v(k)}{\partial a_i(k)}]$$

$$= \eta_{ai}(k)e(k)[\lambda_s (1-u^2(k)) \xi(a_i)(k)], \qquad i=0, 1, 2, \tag{3.28}$$

Using the same as approach as was used in the feedforward weights derivation, $\Delta a_i(k)$, the adjustments in the feedback weights, $\Delta b_j(k)$, are:

$$\Delta b_j(k) = -\eta_{bj} \frac{\partial E(k)}{\partial b_j(k)} = \eta_{bi}(k)e(k)[\lambda_s (1-u^2(k)) \xi(b_i)(k)]. \qquad j=1, 2, \tag{3.29}$$

The term, $\xi_{a_i}(k) = \frac{\partial v(k)}{\partial a_i(k)}$, in Equation (3.28) and the term, $\xi_{b_j}(k) = \frac{\partial v(k)}{\partial b_j(k)}$, in Equation (3.29) are the "sensitivity" signals. The sensitivity signals represent the direct impact of the parameter vector through the system equation on the neural unit response. These sensitivity signals are required in the derivation of the learning and adaptation algorithm.

Substituting Equation (3.26) into the feedforward weight sensitivity signals, $\xi_{a_i}(k) = \frac{\partial v(k)}{\partial a_i(k)}$, the feedforward weight sensitivity signals are found to be:

$$\xi_{a0}(k) = r(k), \qquad \text{for } i = 0, \tag{3.30}$$

$$\xi_{a1}(k) = r(k-1), \qquad \text{for } i = 1, \tag{3.31}$$

$$\xi_{a2}(k) = r(k-2), \qquad \text{for } i = 2 . \tag{3.32}$$

The sensitivity signals for the feedback weights are now considered. Substituting Equation (3.26) into the feedback weight sensitivity signals term, $\xi_{b_j}(k) = \frac{\partial v(k)}{\partial b_j(k)}$, the feedback sensitivity term is obtained as follows:

$$\xi_{g(b_j)}(k) = \frac{\partial v(k)}{\partial g(b_j)}$$

$$= \frac{\partial[a_0(k)r(k) + a_1(k)r(k-1) + a_2(k)r(k-2) - b_1(k)u(k-1) - b_2(k)u(k-2)]}{\partial g(b_j)}$$

$$= -\frac{\partial[b_1 u(k-1)]}{\partial g(b_j)} - \frac{\partial[b_2 u(k-2)]}{\partial g(b_j)}$$

$$= -[\frac{\partial b_1}{\partial g(b_j)}u(k-1) + b_1\frac{\partial[u(k-1)]}{\partial g(b_j)}] - [\frac{\partial b_2}{\partial g(b_j)}u(k-2) + b_2\frac{\partial[u(k-2)]}{\partial g(b_j)}]$$

$$= -\{\frac{\partial b_1}{\partial g(b_j)}u(k-1) + b_1\frac{\partial[u(k-1)]}{\partial[v(k-1)]}\frac{\partial[v(k-1)]}{\partial g(b_j)}\}$$

$$-\{\frac{\partial b_2}{\partial g(b_j)}u(k-2) + b_2\frac{\partial[u(k-2)]}{\partial[v(k-2)]}\frac{\partial[v(k-2)]}{\partial g(b_j)}\}$$

$$= -\{\frac{\partial b_1}{\partial g(b_j)}u(k-1) + b_1\lambda_s\psi'[v(k-1)]\frac{\partial[v(k-1)]}{\partial g(b_j)}\}$$

$$-\{\frac{\partial b_2}{\partial g(b_j)}u(k-2) + b_2\lambda_s\psi'[v(k-2)]\frac{\partial[v(k-2)]}{\partial g(b_j)}\}. \qquad (3.33)$$

For j =1, Equation (3.33) can be written as

$$\xi_{b_1}(k) = \frac{\partial v(k)}{\partial b_1}$$

$$= -(\frac{\partial b_1}{\partial b_1}u(k-1) + b_1\lambda_s\psi'[v(k-1)]\frac{\partial[v(k-1)]}{\partial b_1})$$

$$-(\frac{\partial b_2}{\partial b_1}u(k-2)+b_2\lambda_s\psi'[v(k-2)]\frac{\partial[v(k-2)]}{\partial b_1}). \quad (3.34)$$

As $\xi_{b_1}(k-1)=\frac{\partial v(k-1)}{\partial b_1}$ and $\xi_{b_1}(k-2)=\frac{\partial v(k-2)}{\partial b_1}$, Equation (3.34) becomes:

$$\xi_{b_1}(k)=\frac{\partial v(k)}{\partial b_1}$$

$$=-u(k-1)-b_1\lambda_s\psi'[v(k-1)]\frac{\partial[v(k-1)]}{\partial b_1}-b_2\lambda_s\psi'[v(k-2)]\frac{\partial[v(k-2)]}{\partial b_1}$$

$$=-\psi[v(k-1)]-b_1\lambda_s\psi'[v(k-1)]\xi_{b_1}(k-1)-b_2\lambda_s\psi'[v(k-2)]\xi_{b_1}(k-2). \quad (3.35)$$

For $j=2$, Equation (3.33) can be written as:

$$\xi_{b_1}(k)=\frac{\partial v(k)}{\partial b_2}$$

$$=-(\frac{\partial b_1}{\partial b_2}u(k-1)+b_1\lambda_s\psi'[v(k-1)]\frac{\partial[v(k-1)]}{\partial b_2})$$

$$-(\frac{\partial b_2}{\partial b_2}u(k-2)+b_2\lambda_s\psi'[v(k-2)]\frac{\partial[v(k-2)]}{\partial b_2}). \quad (3.36)$$

Since $\xi_{b_2}(k-1)=\frac{\partial v(k-1)}{\partial b_2}$, and $\xi_{b_2}(k-2)=\frac{\partial v(k-2)}{\partial b_2}$, Equation (3.36) becomes:

$$\xi_{b_2}(k)=\frac{\partial v(k)}{\partial b_2}=-b_1\lambda_s\psi'[v(k-1)]\frac{\partial[v(k-1)]}{\partial b_2}-(\frac{\partial b_2}{\partial b_2}u(k-2)+b_2\lambda_s\psi'[v(k-2)]\frac{\partial[v(k-2)]}{\partial b_2})$$

$$=-\psi[v(k-2)]-b_1\lambda_s\psi'[v(k-1)]\xi_{b2}(k-1)-b_2\lambda_s\psi'[v(k-2)]\xi_{b_2}(k-2). \quad (3.37)$$

Substituting Equations (3.30), (3.31), and (3.32) in to Equation (3.28), and Equations (3.35) and (3.37) into Equation (3.29), the weights update equations (3.21) and (3.22) are completed.

3.3.3 Dynamic Neural Networks (DNNs) using Dynamic Neural Units (DNUs)

A single dynamic neuron can be used to simulate or control single-input, single-output nonlinear systems. However, it is believed that the real power of neural computation comes from neurons connected in a network structure. Larger networks generally offer greater computational capabilities [Hunt, et al., 1992; Hush, et al., 1993; Poggio, et al., 1990]. The multilayer networks in which the neurons are arranged in layers have been proven to have capabilities much more than those of a single layer. The DNUs can be arranged in parallel and in series. There are five DNN structures in the literature [Gupta, et al., 1993 and 1992; Deshpande, et al., 1998; Song, et al., 1999; Song, 2001; Srivastava, et al., 1998], and they are shown in Figures 3.12, 3.13, 3.14, 3.15 and 3.16.

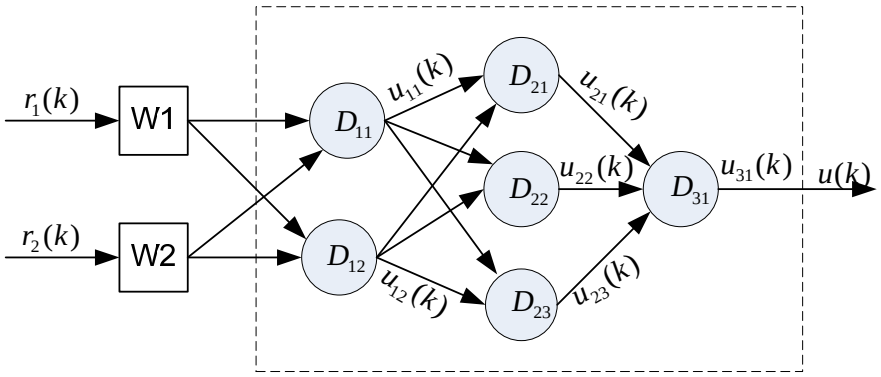


Figure 3.12 Structure of a three stage DNN-1 using six DNUs

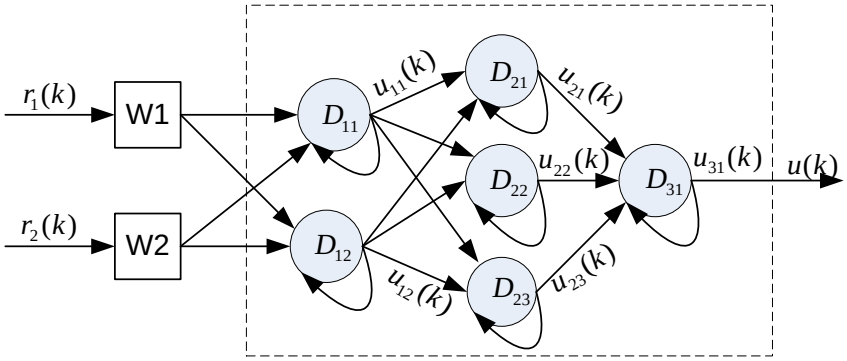


Figure 3.13 Structure of a three stage DNU-2 using six DNUs

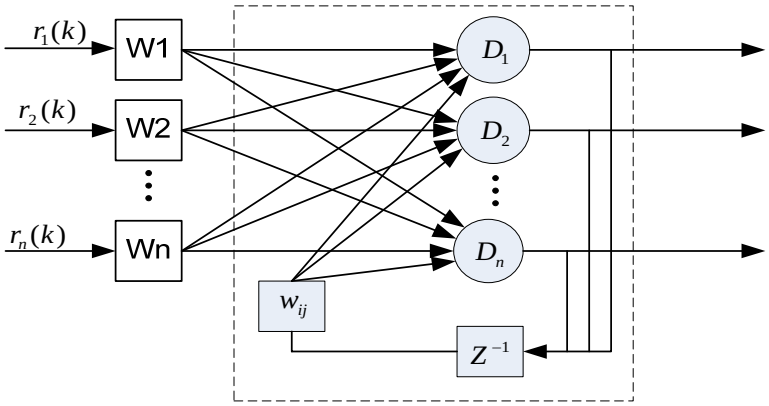


Figure 3.14 Structure of a three stage DNU-3.

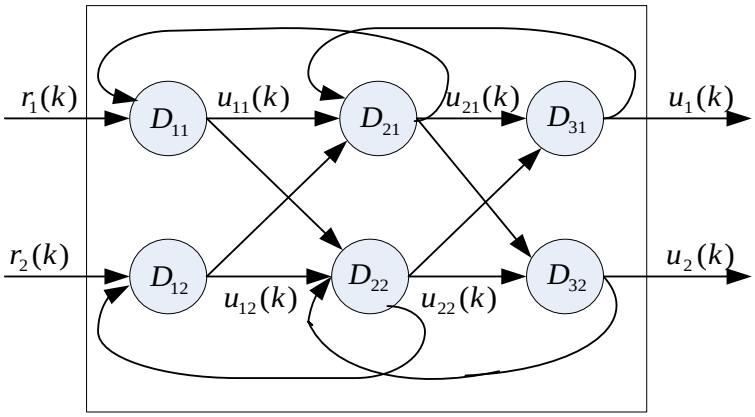


Figure 3.15 Structure of a three stage DNN-4 using six DNUs.

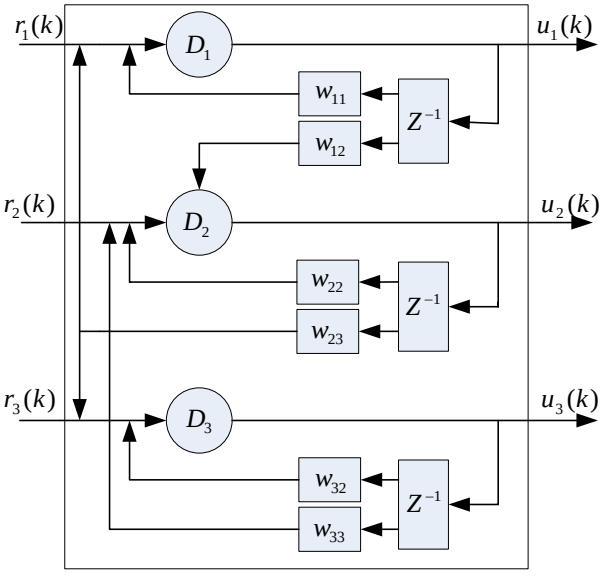


Figure 3.16 Structure of a three stage DNN-5 using three DNUs.

Considering the error accumulation problem in the previous research, which was caused by the external feedback connections in the NN based model, a feedforward structure, DNN-1 was adopted in this present study. Because the order of the "pump only" model adopted in this study was a second order system (details are presented in the next Chapter), and because the DNU-3 was a second order dynamic unit itself, it was not necessary to link the DNUs serially. From a computational requirement point of view, the simpler the structure, the better. As such, a DNN structure which consisted of two DNUs in parallel was employed in this study and is shown in Figure 3.17. In this structure, W_1 and W_2 were defined as "proportional weights" and they were trained by the same algorithm as in internal weights training.

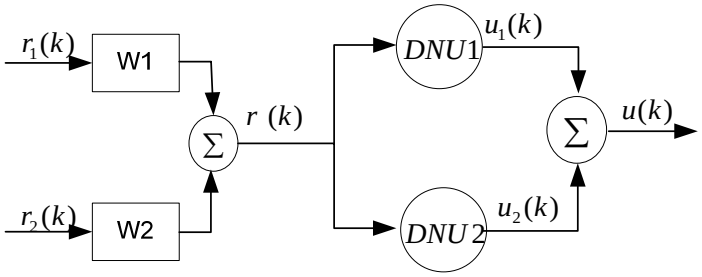


Figure 3.17 Structure of DNN adopted in this study.

3.4 Summary

Several morphologies of the Dynamic Neuron Units were presented in this chapter. The introduction of dynamics into the network makes the training and implementation more complex and the computing more efficient. The gradient descent algorithm was present for the DNU adopted in this study. Several multi-layer structures based on the DNUs presented in this section were also introduced. Based on the order of the pump only model, the DNUs structure which was deemed most suitable for this study was introduced.

CHAPTER 4 FEASIBILITY TEST USING A MATHEMATICAL MODEL OF THE PUMP

4.1 Introduction

In the previous three chapters, studies on hydraulic load sensing systems, modeling of such systems and the neural network approach to modeling were reviewed. It was proposed that the most suitable dynamic neural network morphology for the “pump only” model structure having two inputs and one output was one using two dynamic neural units (DNUs) in parallel. The learning algorithm (based on least mean squares) for the DNN weights updating was also introduced. In this chapter, the “pump only” model (experimentally verified in previous studies) is presented in a “power bond graph” form [Dransfield, P., 1981]. The DNN is trained to “mimic” the dynamic performance of this model by subjecting both the model and the DNN to identical inputs. The DNN is then validated by inputting to both the model and DNN a series of signals which are not used in the training process.

4.2 Mathematical Pump Model

For initial feasibility investigations, it is common practice to use an established model of a pump rather than a “real” pump to train the DNN. This was the approach used in this study. The pump model reflected many of the dynamic nonlinear characteristics of the real pump, and had been verified in other studies [Wright, 1988, Kavanagh, 1987]. If the DNN can be trained to represent a validated simulated model, then it would be capable of representing a real pump from which the mathematical model was established. A second consideration was that if the dynamic

network could not be used to model a “noiseless” analytical model then there would be little chance of this approach being successfully applied to more a complex “real” pump. Therefore, the pump model developed by Wight [1988] and Kavanagh [1987] was adopted in this study.

In the development of the model, several assumptions were made and are:

- The load pressure effect is significant.
- The control piston displacement and its derivatives are linearly related to the swash plate angular displacement and appropriate derivatives.
- Laminar conditions govern the flow in all leakage paths.
- Stiction and viscous friction effects are significant but Coulomb friction is not.
- The mass of the spring, spring cap, yoke, pintle, swash plate, and control piston are lumped into one inertial term as are their damping and friction forces.
- The spring is linear.
- Turbulent conditions govern the flow across all orifices.
- The tank pressure is zero.
- The casing pressure is zero.
- Resistive and capacitive effects are lumped where appropriate.

In order to understand the development of the pump simulation model, it is necessary to understand some of the interactions which exist in the pump.

For normal pump operation, the swash plate is often at full stroke, held against a mechanical stop by the return spring (see Figure 4.1). This creates a condition of maximum flowrate from the pump. However, the output flow is reduced somewhat due to leakage. Oil continually leaks between the pistons and bores, across the valve plate, and through a small hole in the pistons which is used to provide lubrication for the slippers. This oil leaks from the high pressure casing

of the pump to the low pressure case drain and then to tank. For simplicity, the leakage is lumped into one term and is represented by the resistance term R_l.

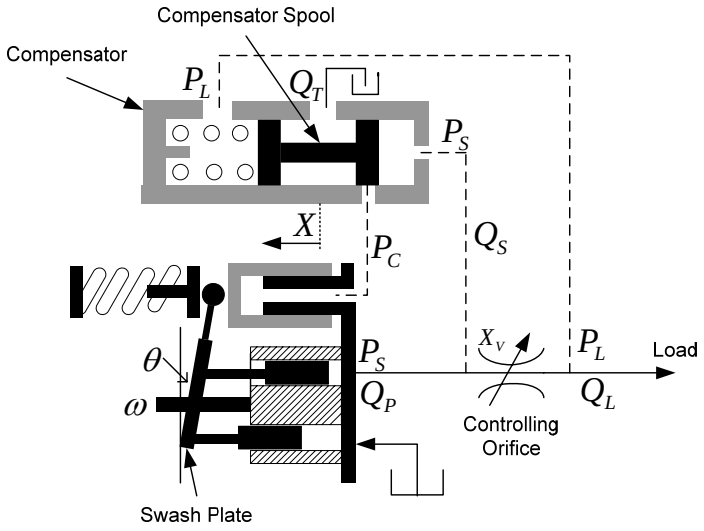


Figure 4.1 Schematic diagram of a load sensing pump

The load in the pump output line provides resistance to the flow of oil. Consequently, pressure is developed as the pump forces oil into the line. This pressure causes a resistance force on the pistons and hence an unbalanced torque on the swash plate. This torque is the effect of load pressure, and it tends to "stroke" the pump (increase the swash plate angle and hence the output flow rate) [Kavanagh, 1987].

To destroke the pump, pressure must be ported from the pump output line to the control piston (see P_S and P_C line in Figure 4.1). The resulting force on the control piston opposes the return spring force, tending to destroke (decrease the swash plate angle) the pump. To stroke the pump, pressure in the control piston chamber fluid is ported to tank via the control compensating piston (compensator spool in Figure 4.1).

The entire swash plate and piston assembly resides in the pump casing which is always full of oil. This oil provides lubrication for sliding parts, but also provides viscous damping which

resists any motion of the swash plate. Acceleration of the swash plate is dictated by the inertia of the swash plate and piston assembly, and the magnitude of the forces applied.

The dynamic equations for the pump simulation model were formulated using the power Bond Graph technique [Dransfield, 1981]. The Bond Graph is a representation of power flow within a system, and as such is a useful tool in arranging the describing equations for hydraulic components and systems. The interdependency of variables can be determined following well established rules for assigning causality between variables. The power Bond Graph of the "pump only" model is shown in Figure 4.2. A brief summary of the Power Bond Graph is presented in Appendix A.

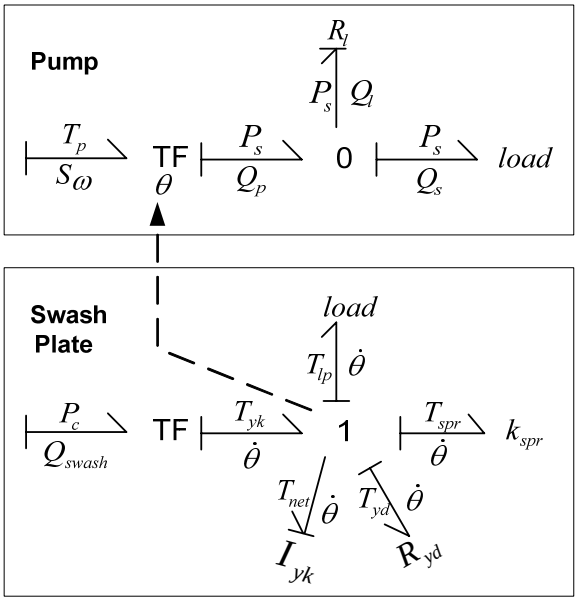


Figure 4.2 Power bond graph of "pump only" model

The describing equations generated from the Bond Graph of Figure 4.2 are:

$$Q_p = (S_\omega)(k_\omega) \cdot \theta \qquad (4.1)$$

$$Q_l = R_l \cdot P_s \qquad (4.2)$$

$$Q_s = Q_p - Q_l \qquad (4.3)$$

$$T_{yk} = P_c \cdot A_{cp} \cdot b\,, \qquad (4.4)$$

$$T_{spr} = k_{spr} \int_0^t \dot{\theta} dt + T_{spr}(0)\,, \qquad (4.5)$$

$$T_{lp} = k_{pr1} + k_{pr2} P_s + k_{pr3} P_s \theta + k_{pr4} \dot{\theta}\,, \qquad (4.6)$$

$$T_{yd} = R_{yd} \dot{\theta}\,, \qquad (4.7)$$

$$T_{net} = T_{yk} - T_{spr} - T_{yd} - T_{lp}\,, \qquad (4.8)$$

$$\dot{\theta} = \frac{1}{I_{yk}} \int_0^t T_{net} dt + \dot{\theta}(0)\,. \qquad (4.9)$$

The values of all parameters and coefficients to be used in the equations above are listed in Appendix B. These equations listed in this causal form readily lend themselves to computer simulation. They were implemented in a Matlab/Simulink environment.

A very important factor in defining the morphology of a DNN is to have an idea of what is the order of the plant. This is not a requirement but it certainly is a useful tool in trying to find a suitable DNN morphology. Thus the choice of the DNU/DNN morphology should demonstrate the same order of the dynamics as the pump model. By using linearization techniques, the transfer function of the "pump only" model was derived from the above Bond Graph equations and it is shown to be:

$$Q_S(s) = \frac{AP_C(s) - (Bs^2 + Cs + D)P_S(s)}{Fs^2 + Gs + H}\,. \qquad (4.10)$$

The details of the derivation and the coefficients values are shown in Appendix C. It is apparent that the "pump only" model contains two poles for P_C and two poles and two zeros for P_S in this linearized transfer function.

4.3 Training Signals Preparation

To create an accurate network model of a component, data used to train the DNN must be generated for all possible behaviour of the components. Thus, an important requirement for defining a set of training signals is that they must excite all operating dominant cut-off frequencies and time constants of the plant in order to obtain full information about input-output properties of that plant. Such an input signal is referred as “persistently exciting” or “general enough” or “rich enough” in the literature [Narendra, et al., 1989]. For dynamic system identification, the requirement of persistent excitation means that the input (training signal) must be sufficiently rich in frequency content and in amplitude variations. In this study, a special input signal was developed based on the pump model cut-off frequency.

Because the DNN must be trained for steady state and transient responses, the training signal (input and output data pairs) should include both steady state and transient information. When the training signal (input and output pairs) frequency is substantially smaller than the “cut-off” frequency or dominant pole/zero of the pump model, the DNN is only trained for the steady state. If the training signal frequency equals to and greater than the dominant “cut-off” frequency of the pump model, the DNN is trained for the transient response of the pump model. Therefore, in order to train the DNN to represent the dynamic characteristics of the pump model over a practical operating range, the training signal should be not only rich in amplitude, but also rich in frequency consisting of several practical operating frequencies.

The uniformly distributed random signal in Matlab/Simulink was adopted to achieve the richness in amplitude. The cut-off frequency is a critical point from the training signal frequency selection standpoint and it can be obtained from the system frequency response. However, the frequency response of a two input system such as the “pump only” model cannot be obtained if both the two inputs vary simultaneously. However, the two input system frequency can be

obtained from the frequency response where one input is fixed as constant. Figure 4.3 shows the pump magnitude frequency response at P_S equals to $17.2MPa$ and Figure 4.4 shows the pump magnitude frequency response at P_C equals to $3.1MPa$.

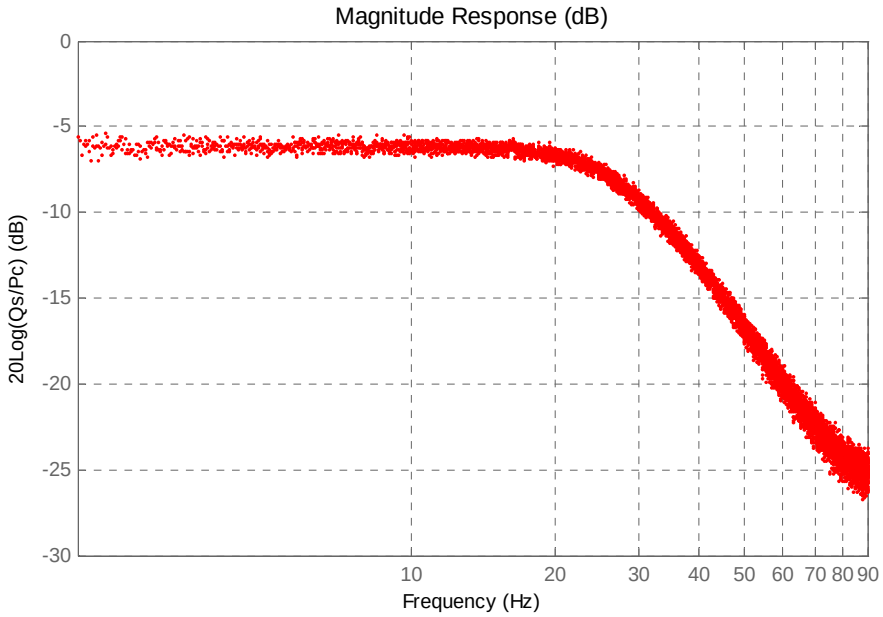


Figure 4.3 Magnitude frequency response of "pump only" model ($P_S = 17.2MPa$).

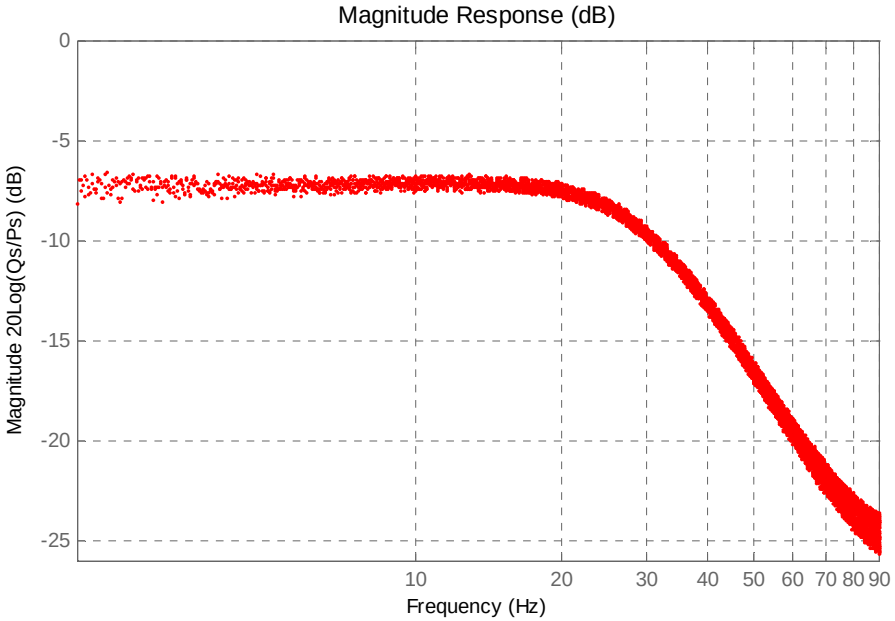


Figure 4.4 Magnitude frequency response of "pump only" model ($P_C = 3.1MPa$).

From Figure 4.3, it is evident that the "pump only" model is a second order system when P_S is held constant and this is consistent with Equation 4.10. Equation 4.10 also indicates that the "pump only" model is a second order system with two poles and two zeroes of P_S when P_C is set constant. Since the frequency response in Figure 4.4 is very similar to than shown in Figure 4.3 (other than a slight magnitude shift), it is apparent that the numerical values of the two zeroes of P_S in the linearized system transfer function (P_C held constant), are much larger than the dominant poles. The zeroes do not affect the system properties in a frequency range less than 100Hz, a range which was defined as the upper frequency of the "pump only" model.

In Figures 4.3 and 4.4, the cut-off frequency of the pump only model is around 25Hz. From a practical point of view, four times the cut-off frequency, 100Hz, was believed to be a reasonable operating frequency band. Therefore, the training signal consisted of several frequencies over this range. In this present study, five frequencies were chosen to make up the training signal. They were 5Hz, 20Hz, 25Hz, 33Hz, and 100Hz. Training signals of 5Hz and 20Hz were used to train the DNN for steady state information, and training signals of 25Hz, 33Hz, and 100Hz used to train the DNN for dynamic information. To ensure the condition of richness in amplitude, the input P_S was randomly chosen from $0MPa$ to $17.2MPa$, and P_C randomly chosen from $2.8MPa$ to $4MPa$. It should be noted that the range of P_C was chosen to ensure that the swashplate angle of the pump did not saturate at its limits very often. In addition, the chosen amplitude ranges of inputs P_S and P_C limited the output flowrate, Q_S , to be in the range of $0m^3/\sec$ to $3.2\times10^{-4}m^3/\sec$.

Because the outputs of the nonlinear functions of DNN were bounded within -1 to 1, the DNN could not be trained using the real input/output data pairs. Hence, it was necessary to

normalize all the signals of the "pump only" model according to requirements of the DNN. This was accomplished by pre-processing and post processing the data pairs from actual values (MPa, $m^3/\sec$) to normalized values (-1, +1). This normalizing had another advantages. Signals generated by the computer software were also in the -1 to +1 range, and hence were in a very convenient form for training and testing.

Typical "normalized" training input signals for one cycle of P_S, and P_C, and the output normalized flowrate Q_S are shown in Figures 4.5, 4.6, and 4.7. The input signals consisted of steps at different frequencies (rather than sinusoids) because step inputs are inherently very rich in higher frequency content.

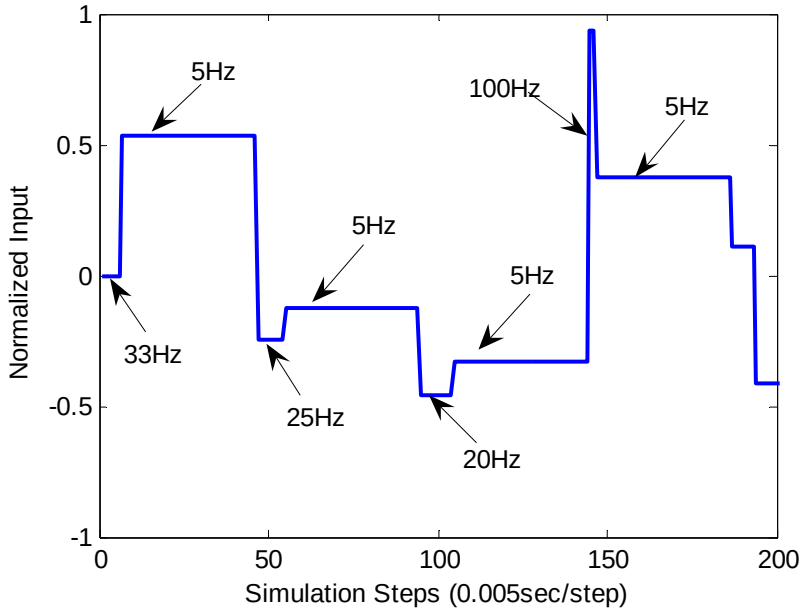


Figure 4.5 One cycle of the normalized training signal, P_S.

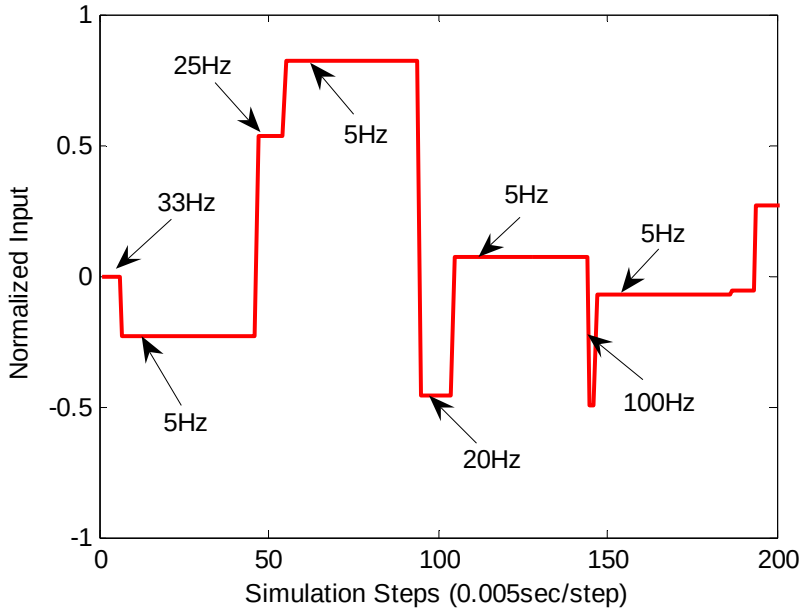


Figure 4.6 One cycle of the normalized training signal, P_C.

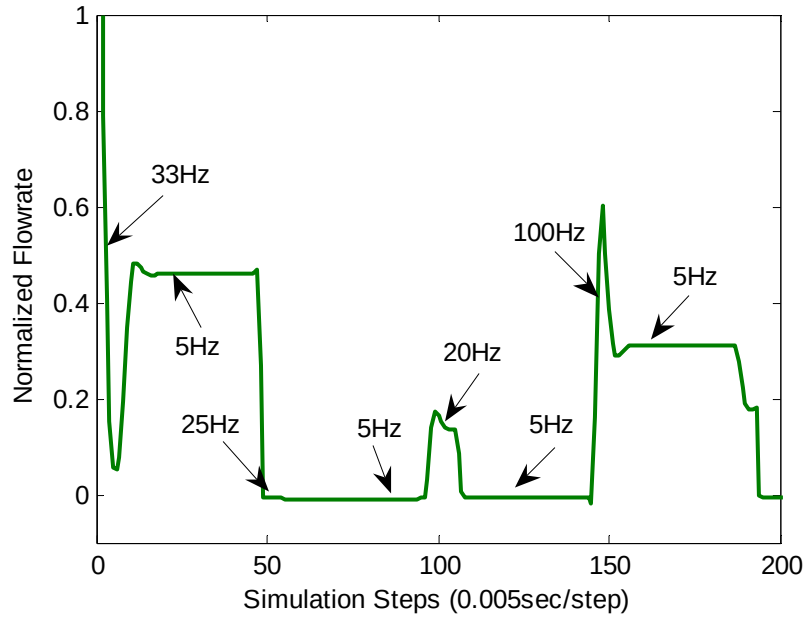


Figure 4.7 One cycle of the normalized training signal, Q_S.

In Figures 4.5, 4.6, and 4.7, the x-axis is given in "Simulation Steps" where one step represents 0.005 seconds. One cycle (0.93 second) of training signal was comprised of four 5Hz

signals, one 20Hz signal, one 25Hz signal, one 33Hz signal and one 100Hz signal; In addition the amplitudes were change randomly (generated form the uniformly distributed random signal and some logical controller in the Matlab/Simulink environment). Figure 4.8 shows some sections of the normalized training signal taken from the whole training process.

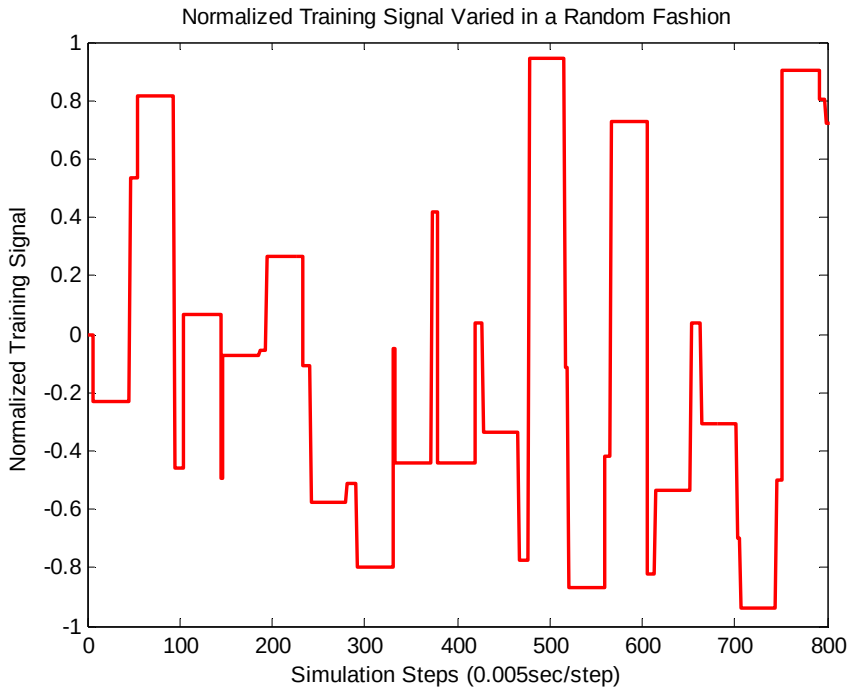


Figure 4.8 Normalized training signal P_C varied in a random fashion.

For the pump model of Wright [1988], it should be noted that the swash plate angle can never operate below an angle of zero radians and above the maximum angle of 0.3125 radians. Therefore, the resulting flowrate limitation was considered in developing the pump model. The model treated any swash plate angle of lower than zero as zero, and swash plate angles greater than 0.3125 radians as 0.3125 radians. These limits were implemented by applying various logic operators to the pump model in the Matlab/Simulink environment. One of these limitations can be observed in Figure 4.7 in the signal around 60s-90s. At this point, the value of P_C and the

value of P_S would result in a the pump model output flow of less then zero, but with the limits set by the logic operator, the flowrate is set to zero.

In summary, the training signals P_S, P_C, and Q_S using random amplitudes and several different operating frequencies of step inputs, were used to train the DNN.

4.4 Model Validity Testing

The DNU based DNN was trained using the gradient descent algorithm for normalized random amplitude signals which contained five different frequencies. All training was considered to be "instantaneous" (see Section 3.1); the sampling time was set to 0.005 seconds. After 20000 seconds, the training error did not decrease further and the training was stopped. At this point, the average value of each weight in the DNN over their last 5000 points was adopted as the final weight value.

In Chapter Three, it was stated that the "trained" DNN must be validated by testing after the training stops. During the testing process, the trained DNN was subjected to some new inputs which were not necessarily used in the training process. The trained DNN output and the model output for these new inputs were compared to see if the trained DNN represented the physical system.

In the next section, the results of the test procedure for various inputs are considered. Note that discussion of the results is deferred to Section 4.5 where all of the trends are correlated and some basic conclusions drawn. These conclusions are required in order to justify the compensation scheme introduced in Chapter 5.

4.4.1 Sine Wave Test

Because of the presence of the two inputs, testing of the trained DNN became more complex. The first series of tests was to subject the DNN to an extreme situation where P_C was a sinusoidal signal and P_S a constant; P_S was then made a sinusoidal signal and P_C held constant. Although from a practical point of view, this situation would never happen in a load sensing system (where P_S and P_C are both frequency and amplitude related due to the interaction of the system). This sort of extreme condition can only occasionally happen in a variable displacement pump where P_S and P_C are truly independent. Subjecting the trained DNN to this extreme operating condition was essential to investigate the performance of the DNN in an extended application range of a variable displacement pump with single input.

Figures 4.9, 4.10 and 4.11 show the DNN output and pump model output comparison results for P_S constant ($16.4MPa$) and P_C a sinusoidal signal (amplitude from $2.8MPa$ to $4MPa$). Figures 4.12, 4.13 and 4.14 show the DNN output and pump model output comparison results for P_C constant ($3.1MPa$) and P_S a sinusoidal input (amplitude from $4.3MPa$ to $12.9MPa$). Note that the output of the DNN has been post-processed to determine the flow in $m^3/\sec$ in these figures.

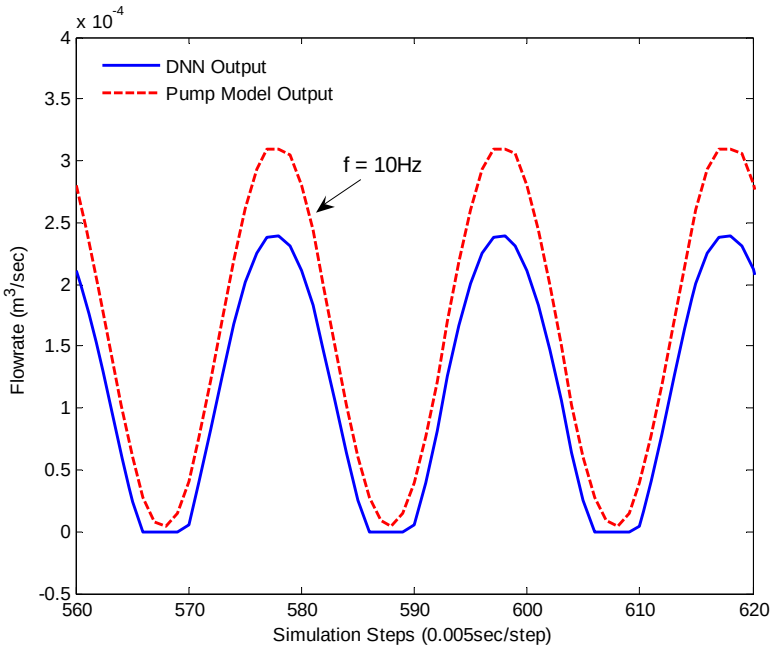


Figure 4.9 Test results for P_S constant ($16.4MPa$) and P_C a sinusoidal input at 10Hz.

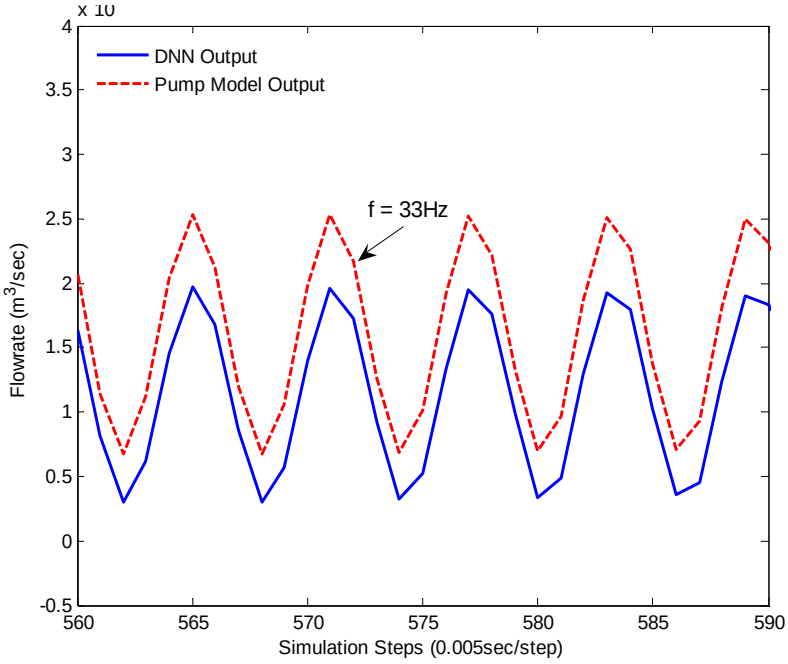


Figure 4.10 Test results for P_S constant ($16.4MPa$) and P_C a sinusoidal input at 33Hz.

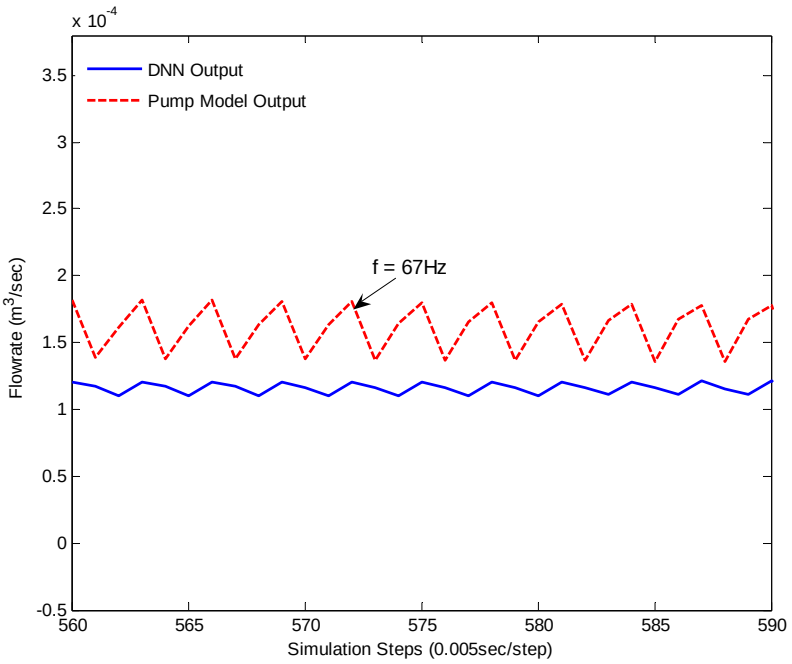


Figure 4.11 Test results for P_S constant ($16.4MPa$) and P_C a sinusoidal input at 67Hz.

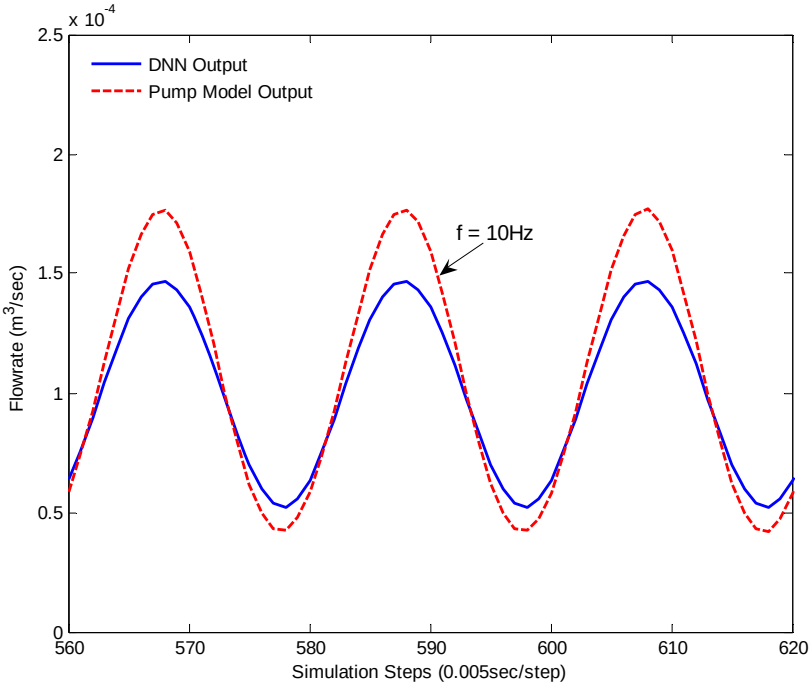


Figure 4.12 Results for P_C constant ($3.1MPa$) and P_S a sinusoidal input at 10Hz.

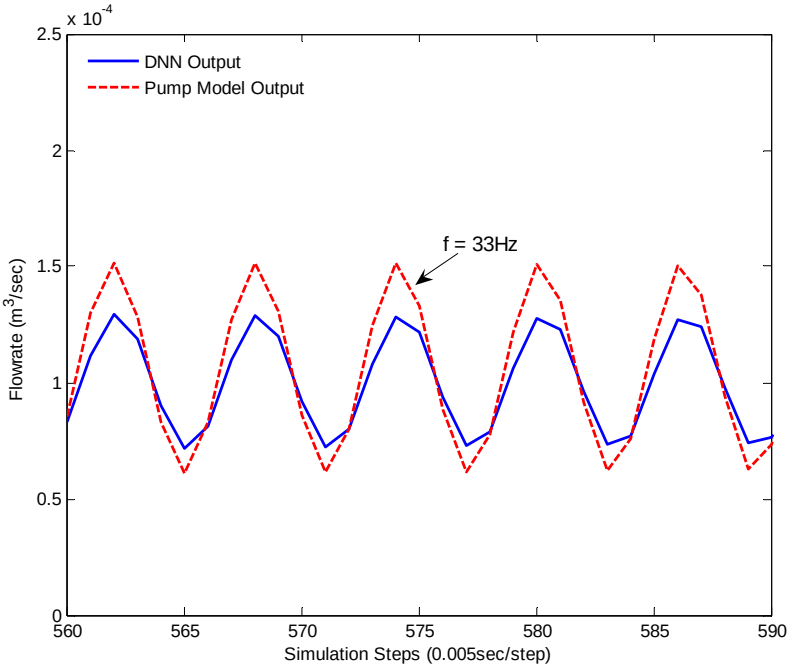


Figure 4.13 Results for P_C constant ($3.1MPa$) and P_S a sinusoidal input at 33Hz.

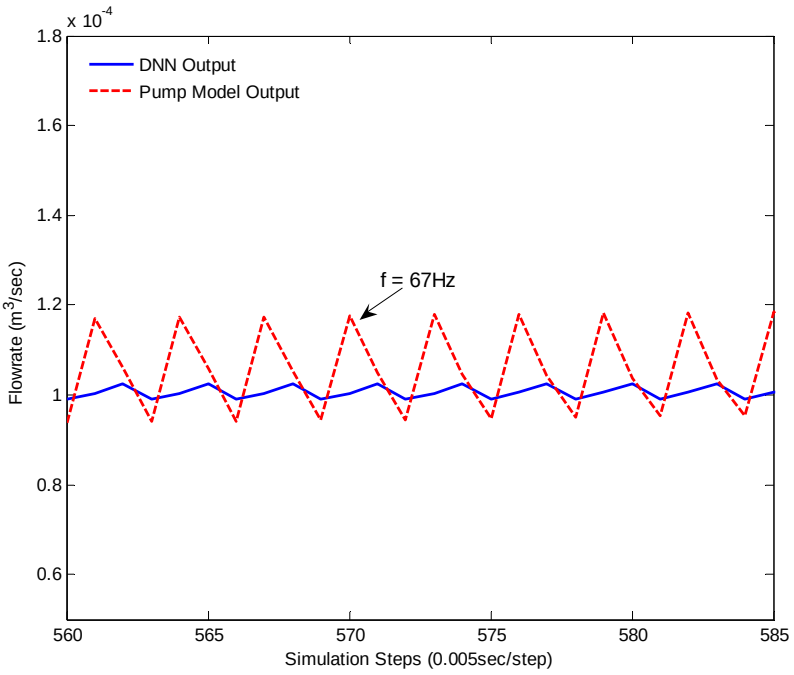


Figure 4.14 Results for P_C constant ($3.1MPa$) and P_S a sinusoidal input at 67Hz.

To demonstrate these results in another form, a frequency response of the trained DNN model was conducted using a "chirp signal function" in Matlab (a swept sinusoidal signal). In these tests, P_S was held constant at $17.2MPa$ and P_C varied in frequency with its upper and lower amplitude values at $4MPa$ and $2.8MPa$ respectively. The frequency response for both the DNN and pump model are shown in Figure 4.15. Also, the frequency responses of the trained DNN and pump model where P_C was fixed at $3.1MPa$ and P_S, a sine wave signal (amplitude varying from $4.3MPa$ to $12.9MPa$) are shown in Figure 4.16.

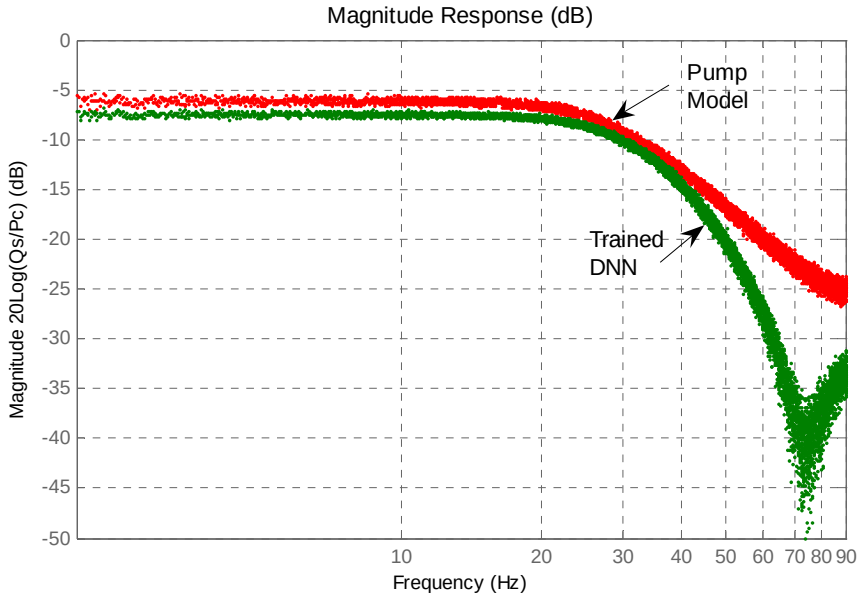


Figure 4.15 Magnitude frequency responses for trained DNN and pump model (P_S =17.2MPa).

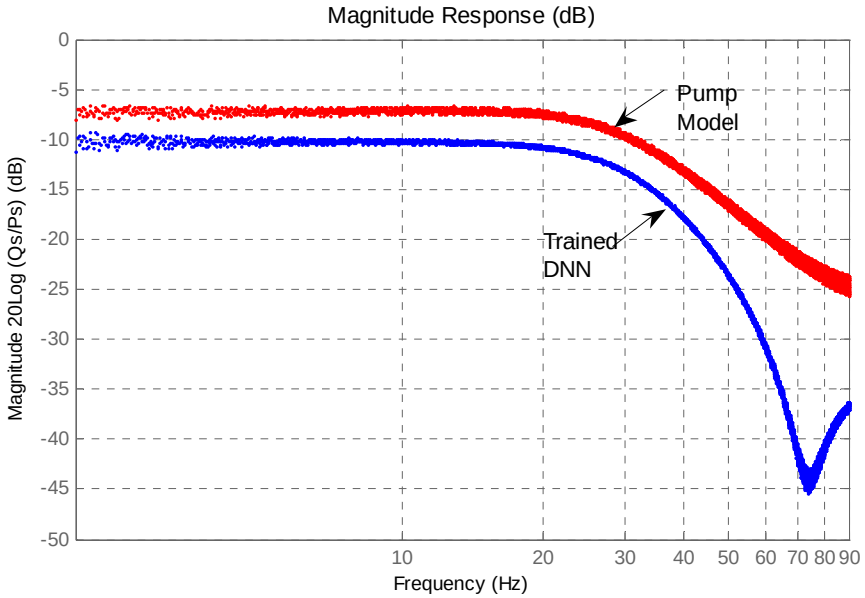


Figure 4.16 Magnitude frequency responses for trained DNN and pump model ($P_C = 3.1MPa$).

As stated above, in a real application, P_S and P_C are not completely independent. They are related in both amplitude and frequency and vary in a similar fashion. Therefore, in the second series of tests, both P_S and P_C are sinusoidal signals which have the same frequency but unrelated random amplitude. As such, the normalized P_S was to be $(A \cdot \sin(2\pi ft))$ and the normalized P_C to $(B \cdot \sin(2\pi ft)$ where A and B were random numbers within -1 and 1. These inputs signals are shown in Figures 4.17 (a), 4.18 (a), 4.19 (a), and 4.20 (a). In order to investigate DNN performance in the whole training range, tests at four frequencies (10Hz, 25HZ, 50 Hz and 100Hz) were examined. The test output results are shown in Figures 4.17 (b), 4.18 (b), 4.19 (b), and 4.20 (b).

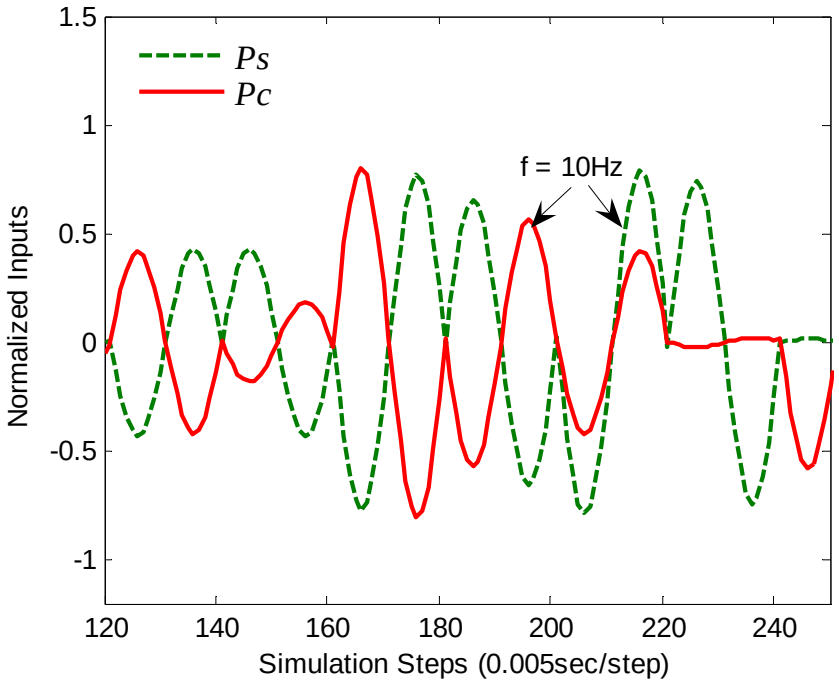


Figure 4.17(a) Both P_S and P_C random sinusoidal inputs at 10Hz .

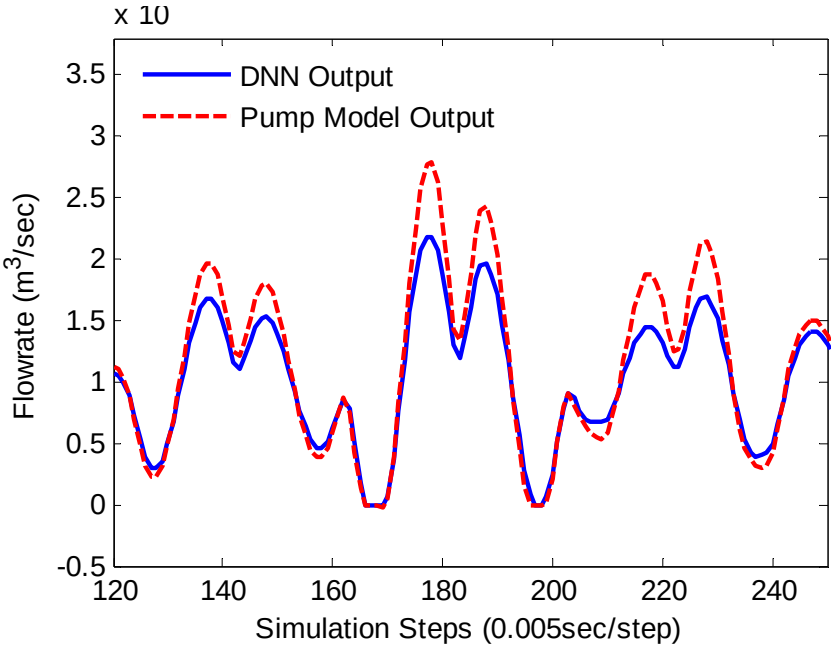


Figure 4.17(b) Test results for input signals shown in Figure 4.17 (a).

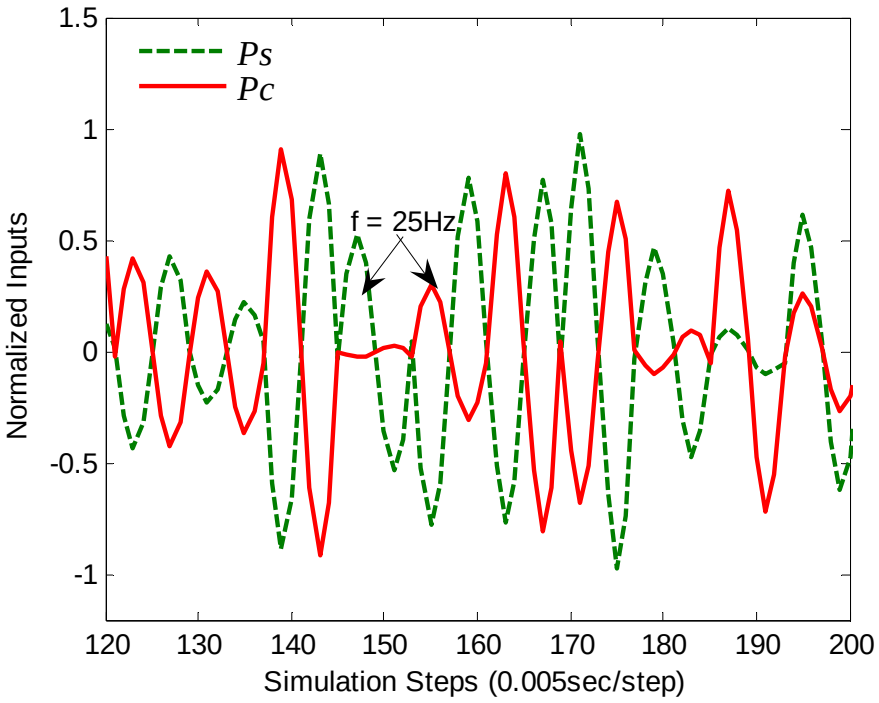


Figure 4.18(a) Both P_S and P_C random sinusoidal inputs at 25Hz.

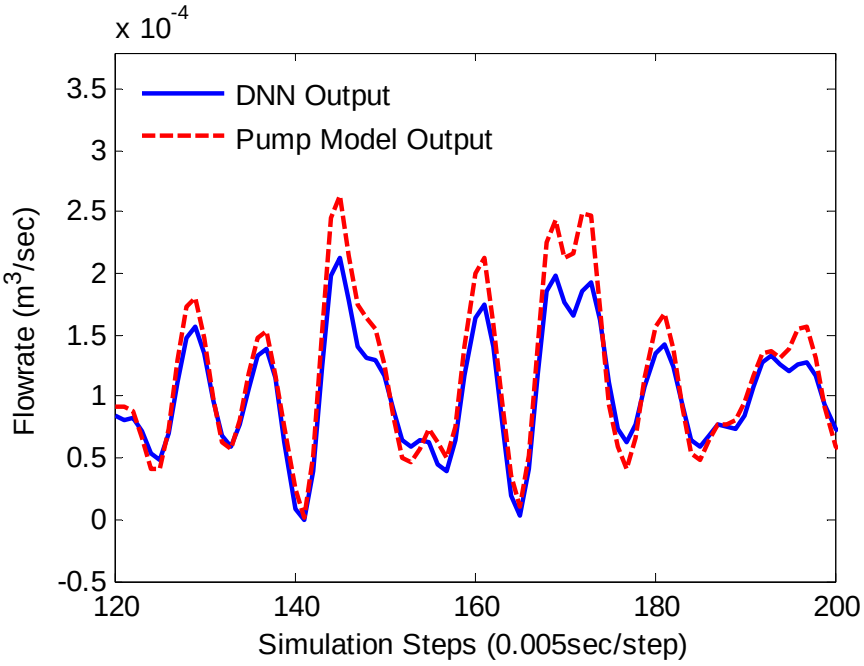


Figure 4.18(b) Test results for input signals shown in Figure 4.18 (a).

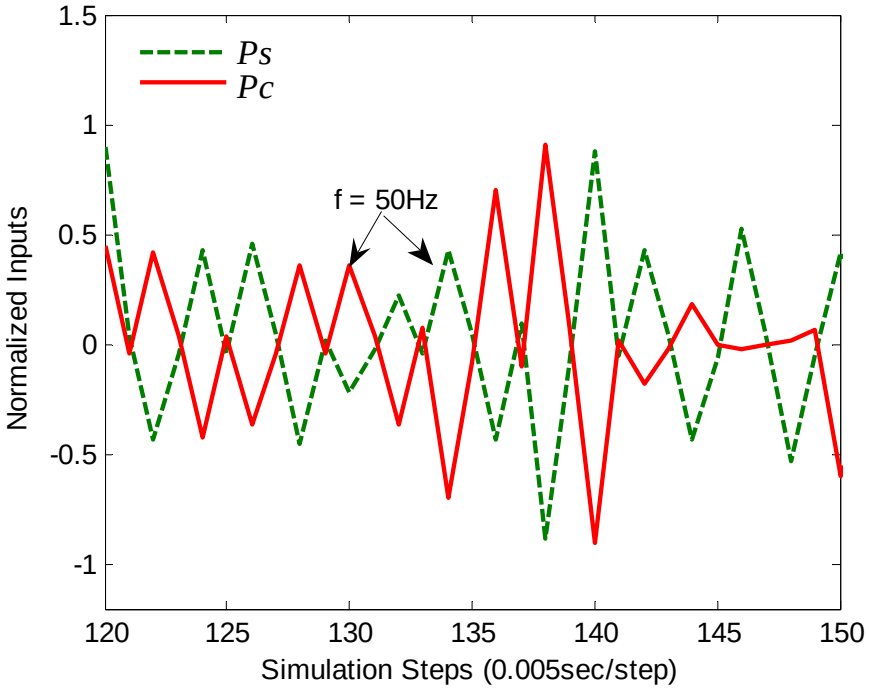


Figure 4.19(a) Both P_S and P_C random sinusoidal inputs at 50Hz.

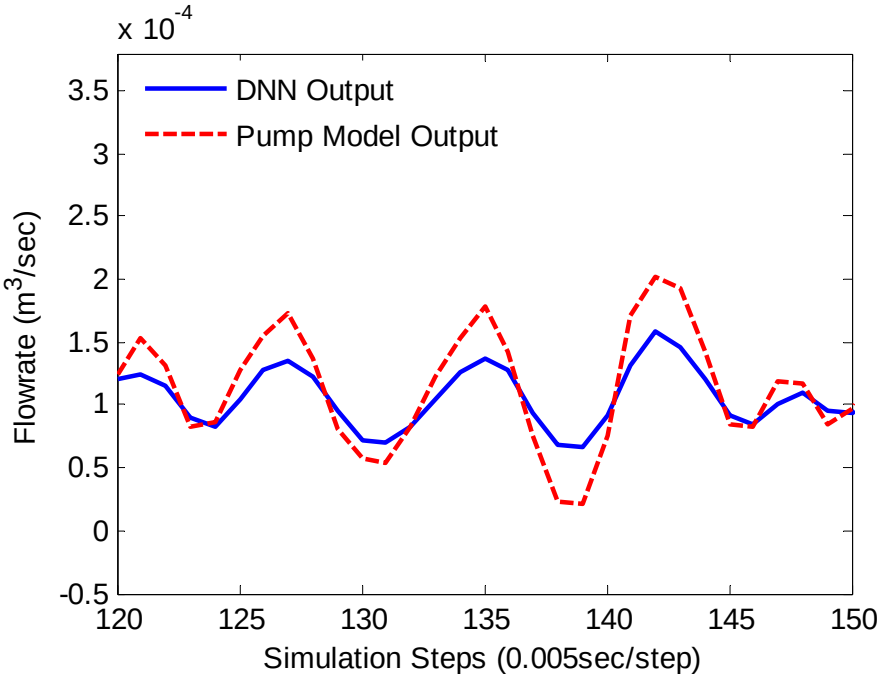


Figure 4.19 (b) Test results for input signals shown in Figure 4.19 (b).

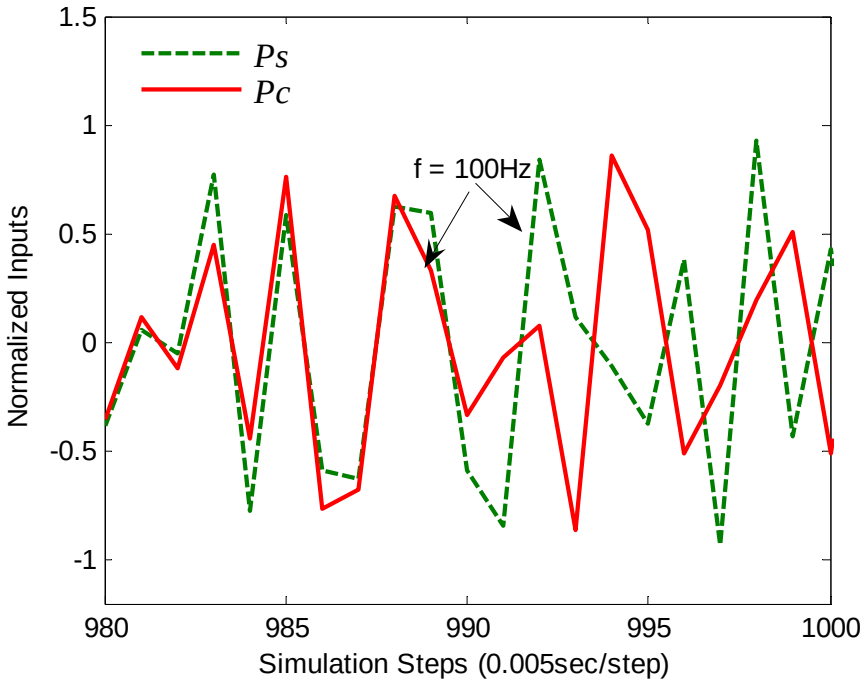


Figure 4.20(a) Both P_S and P_C random sinusoidal inputs at 100Hz.

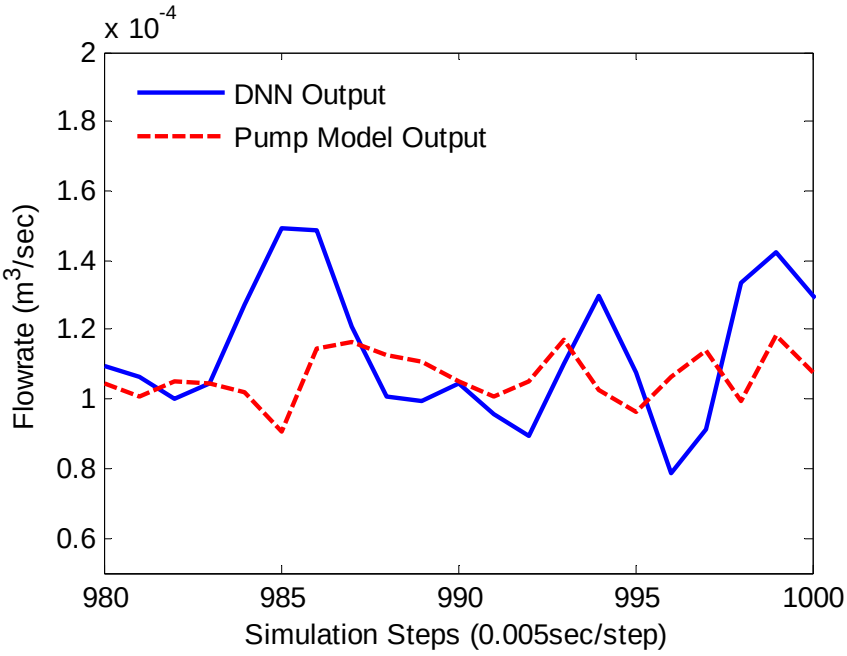


Figure 4.20(b) Test results for input signals shown in Figure 4.20 (a).

4.4.2 Step Input Tests

To demonstrate the validity of the DNN model for step inputs (considered a more appropriate input showing both transient and steady state information in the time domain rather than the frequency domain), both the DNN and the pump model were subjected to a series of controlled step inputs. In the first case, P_C was held constant at $3.4MPa$ and P_S set as a random step input with a frequency of 10Hz. Test results are shown in Figure 4.21. In the second case, P_S was set as $17.2MPa$ and P_C as a random step input with frequency of 10Hz. Test results are shown in Figure 4.22. Another test for the second type of inputs was also performed where both P_S and P_C were random step inputs with the same frequencies of 10Hz. The test results are shown in Figure 4.23.

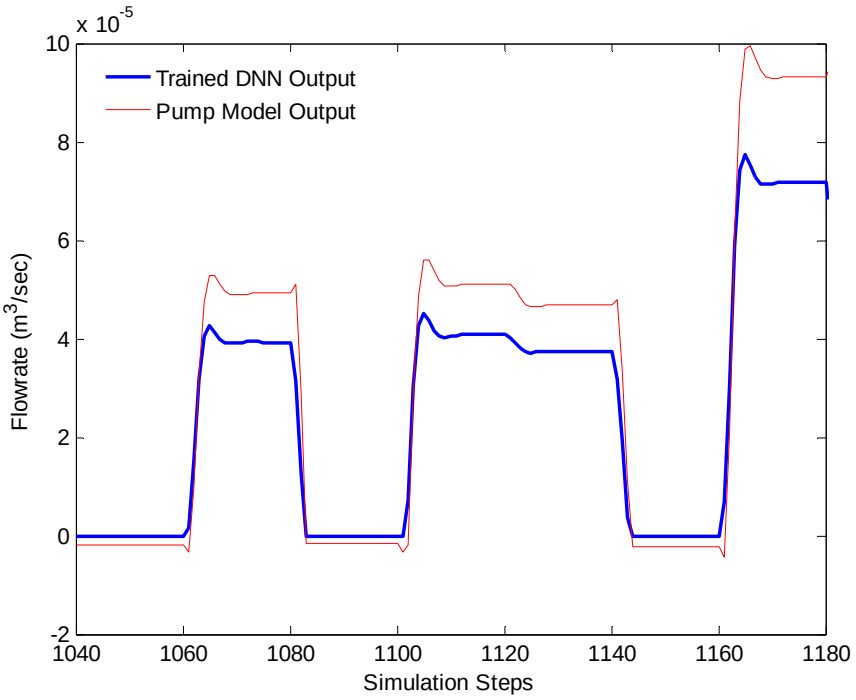


Figure 4.21 Test results for P_C constant ($3.4MPa$) P_S in random step fashion.

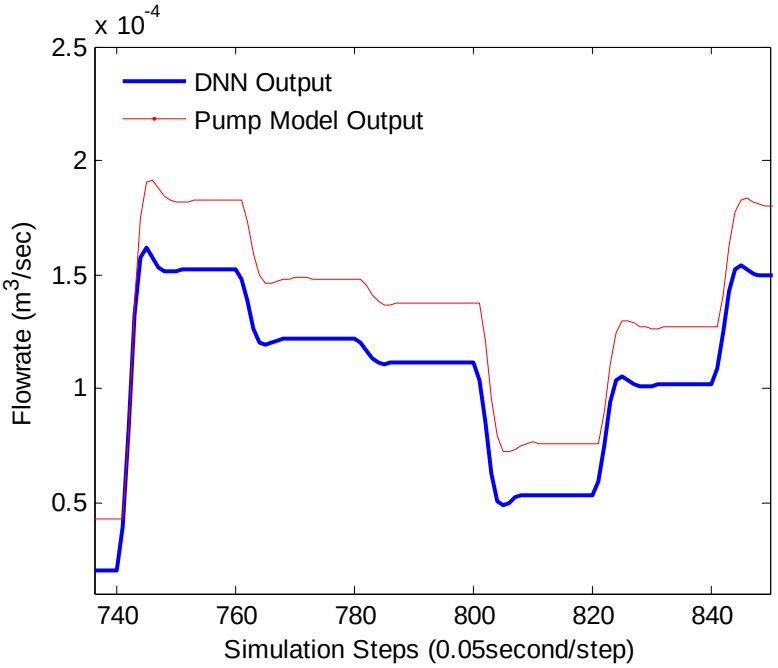


Figure 4.22 Test results for P_S constant ($17.2MPa$) P_C in random step fashion.

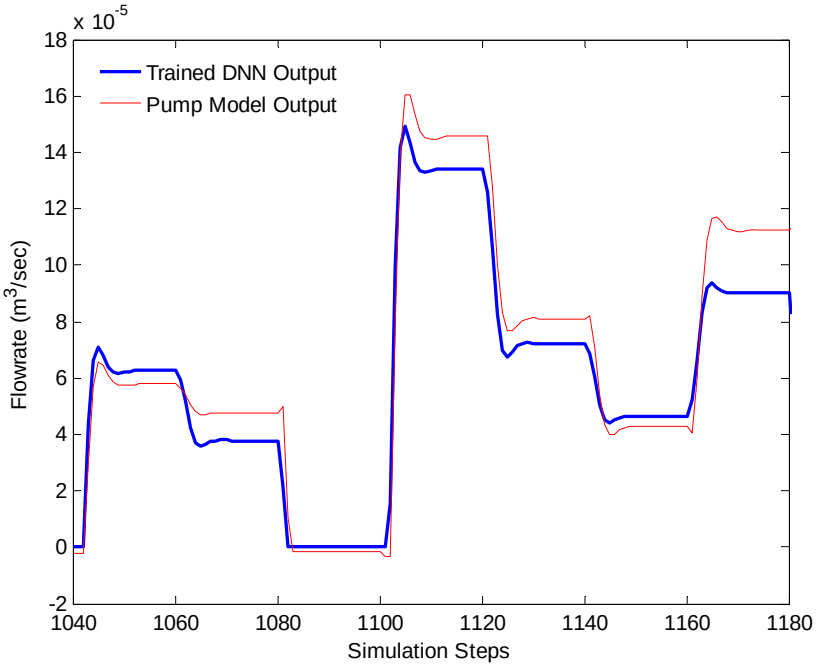


Figure 4.23 Test results for when both P_S and P_C are random step inputs.

4.5 Discussion and Conclusion on Initial Tests:

An observation that can be made was the fact that the error accumulation problem discussed in Section 3.1.2 did not occur. Thus the morphology of the DNN appears to have solved the problem encountered by Xu [1997(1)].

Both the sine wave test results (Figures 4.9-4.20) and the step input test results (Figures 4.21-4.23) indicate that the trained DNN is able to approximate the dynamics of the pump flow output over a frequency range up to approximately 40 Hz, but it is also apparent that the steady state values (using step inputs) and the amplitudes (using sine wave) are substantially different. This is also evident in the "bias" or shift in the frequency response amplitude ratio. In Figures 4.15 and 4.16, the trained DNN frequency response breaks down around 40Hz which is consistent with Figures 4.9 - 4.14; indeed, the order of the pump model and DNN deviate substantially at that point. However, the agreement is acceptable over the dominant break frequencies. A further discussion on the model quality is now necessary to analyze what has affected or has limited the modeling accuracy.

4.5.1 Low Accuracy at Steady State

As mentioned above, the DNN does capture the dynamics of the pump model over the training frequency range (less than 40 Hz). But the steady state error is quite large. A possible reason for this is now forwarded. It was believed that the source of the error could be traced to some of the inherent properties of the DNN structure adopted in this study. A visual inspection of the structure of the DNN (shown in Figure 3.17) shows that the weights, W_k $(k = 0,1,2)$, determine the proportion of P_S and P_C which are input into the nonlinear function input $r(k)$. Since they are internal to the dynamic unit DNU, weights, a_{ni} and b_{nj} determine the dynamics of the DNN. (In this notation, n represents the number of the neuron, n=1.2, i represents the

number of the feedforward weights, i=0, 1, 2 and j represents the number of the feedback weights, j=1, 2). In comparison, the "proportional weights", W_k ($k = 0,1,2$), affect the system steady state.

In order to investigate the steady state error, the last 5000 values of each weight in the training process were examined and some of them are shown in Figures 4.24 (a) to Figure 4.24 (h). For a trained DNN, all the weights should converge to a constant number or vary in a very small range after the training stops. The DNU internal feedforward and feedback weights shown in Figures 4.24(d) to 4.24(h) vary over a small range (less than 10% of the average value), but the DNU "proportional" weights W_k shown in Figure 4.24(a), 4.24(b) and 4.24(c) change significantly (50% of the average value). That means the pump model steady state behaviour has not been "emulated" or "learned" by the DNN as these weights did not converge. On the other hand, since the variations in a_{ni} and b_{nj} are fairly small over the last steps, the internal feedforward and feedback weights can be treated as converged and hence, the dynamic characteristics of the pump model are captured by the DNN.

It was concluded that for the input signals that were used to train the DNN, the DNN could only be trained to capture the pump model dynamics but could only "follow" the steady state data as oppose to "training" to it. This conclusion CANNOT be generalized to all conditions since an extensive experimental and theoretical study could not be conducted in the time frame of this research.

It should be noted that in previous studies by [Gupta, et al., 1993 and 1992; Deshpande, et al., 1998; Song, et al., 1999; Song, 2001; Srivastava, et al., 1998], the DNUs/DNN were used for control purposes where the physical plant was not separated from the DNN. The error signal which was used for weight updating was always available for the continuous training process.

For such control purpose, it is not essential for the DNN to "follow" the steady state or to "learn" the steady state as long as the DNN can produce a right control signal to the plant. However, in the system modeling approach, the ability of "learning" rather than "following" is significant because the trained model (DNN) must display characteristics to the plant and be able to reproduce the performance of the plant under all operating conditions and for any time duration.

Further investigation on the DNU structure and the DNN configuration is needed to overcome the non-convergence "proportion weights" problem. Alternatively, compensating for the changing in "proportion weights" externally might be an effective method to eliminate the steady state error. Such an approach is presented in the next chapter.

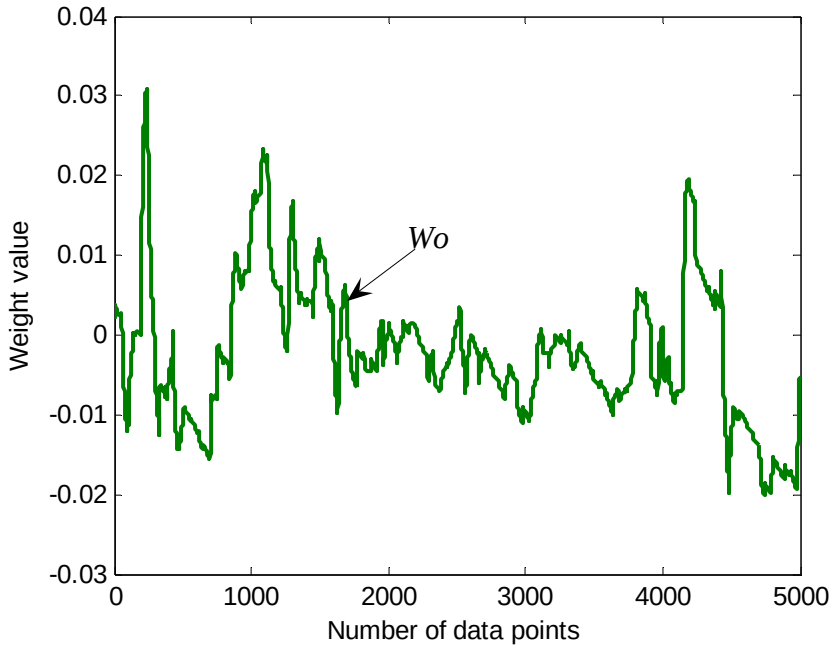


Figure 4.24 (a) The last 5000 weight value of W_0 in training process.

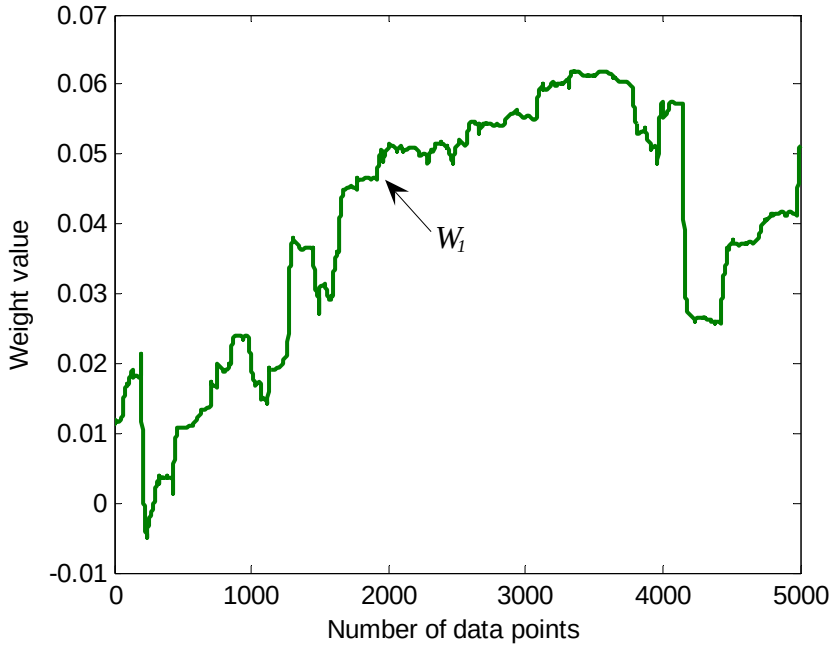


Figure 4.24(b) The last 5000 weight value of W_1 in training process.

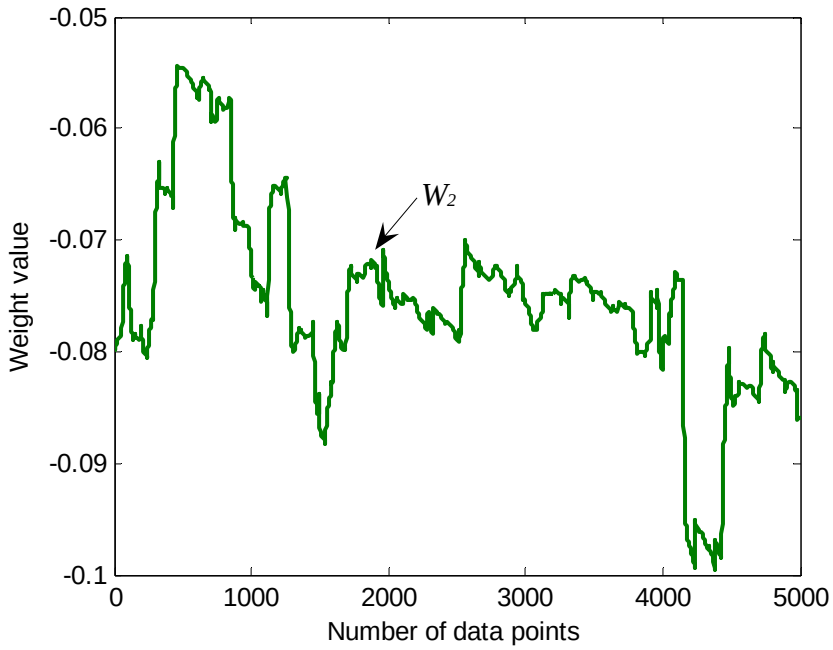


Figure 4.24(c) The last 5000 weight value of W_2 in training process.

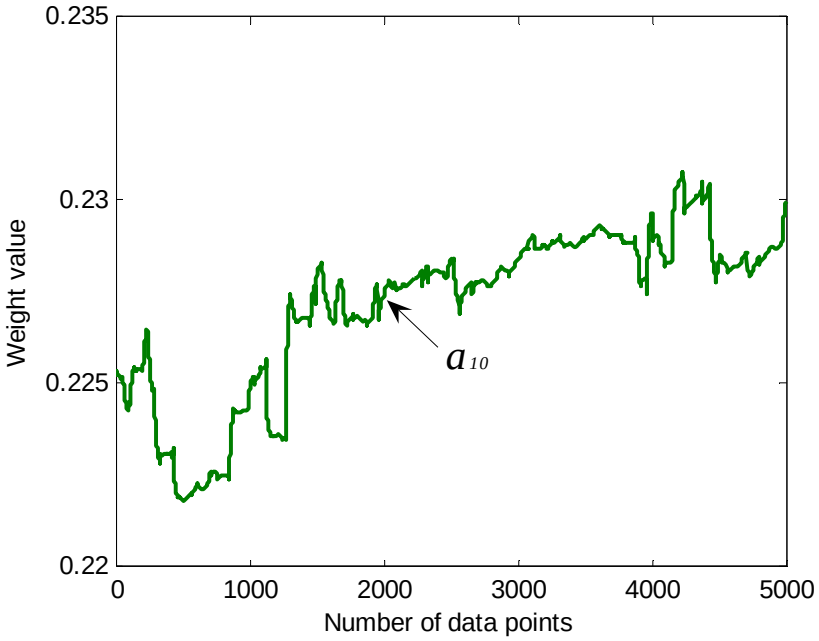


Figure 4.24 (d) The last 5000 weight value of a_{10} in training process.

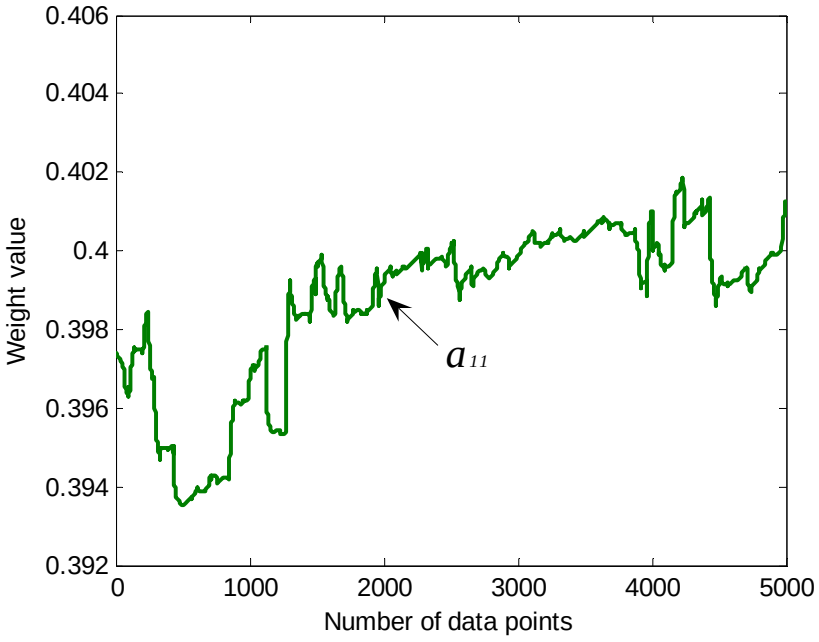


Figure 4.24 (e) The last weight value of a_{11} in training process.

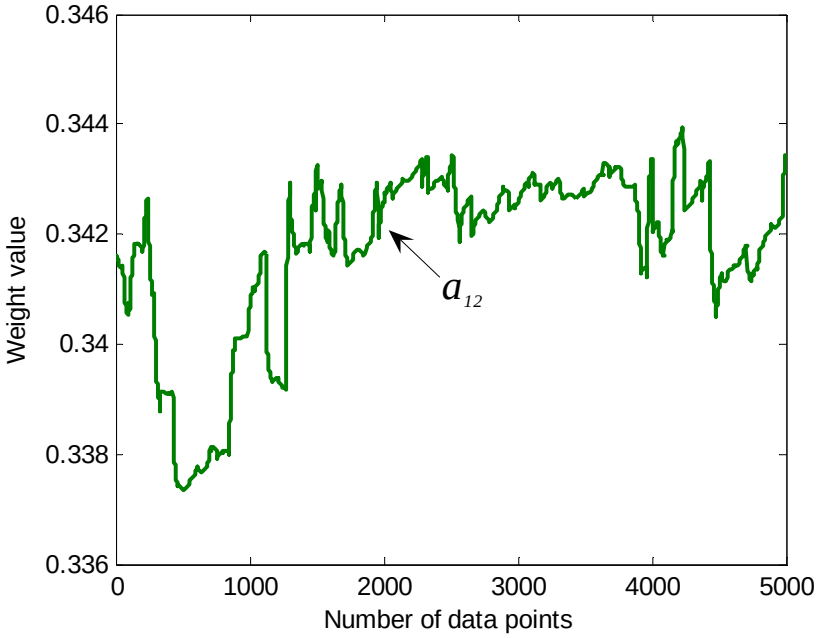


Figure 4.24 (f) The last weight value of a_{12} in training process.

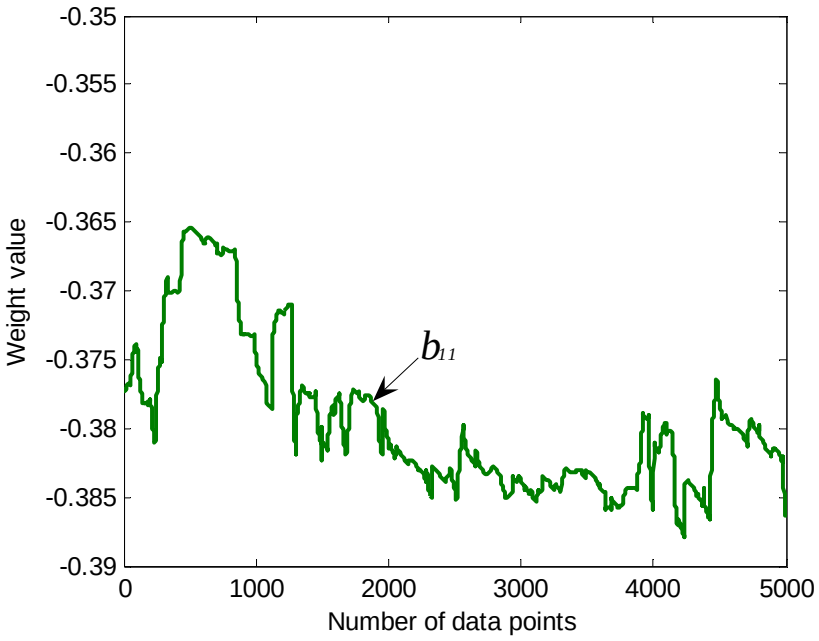


Figure 4.24 (g) The last 5000 weight value of b_{11} in training process.

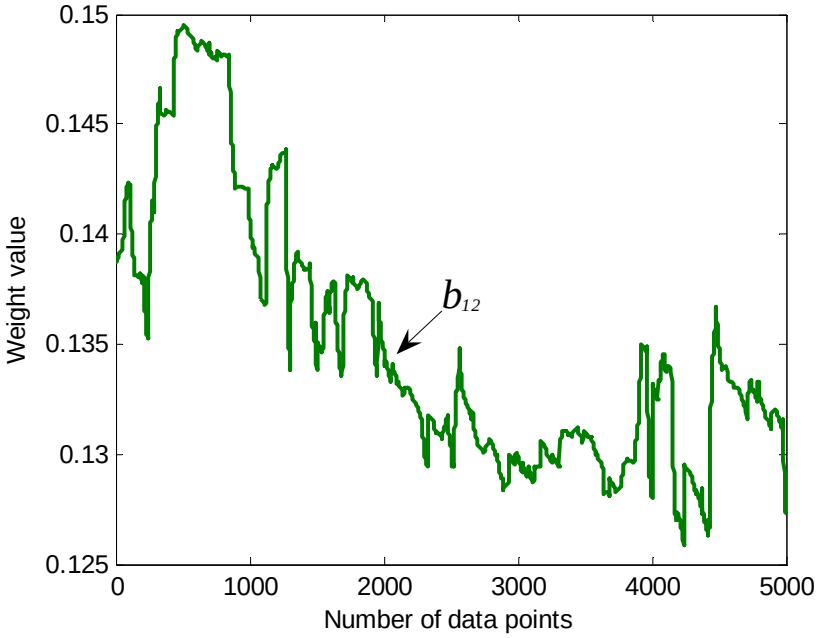


Figure 4.24(h) The last 5000 weight value of b_{12} in training process.

4.5.2 Low Accuracy at High Frequency

Consider Figure 4.15 and Figure 4.16, the DNN frequency response and the pump model frequency response deviate at around 40Hz. The decrease in accuracy is believed to be caused by the lower "richness" in signals greater than 40Hz. The training signal frequencies in one period were four 5Hz, one 20Hz, one 25Hz, one 33Hz and one 100Hz. There were more signals in low (four 5Hz), and middle frequencies (20Hz, 25Hz, and 33Hz), but only one at high frequencies (at 100Hz). There was no input in the range of 40Hz to 100Hz which was used to train the DNN. This explains the higher accuracy around the cut-off frequency (25Hz to 35Hz). This problem can be overcome by adding more different components into the training signal increasing the frequency richness of the training signal.

4.6 Summary

In this Chapter, a mathematical pump model was developed using the Power Bond Graph technique (verified experimentally in other studies). To achieve the richness in both frequency and amplitude of the training signal and considering the cut-off frequency of the pump model, a special training signal comprised of several different frequencies was created in a Matlab environment. Tests to validate the trained DNN model were performed. The results indicated that the trained DNN did capture the nonlinear dynamics of the pump model, but the steady state error was quite large. The last 5000 values of the weights were examined, and it was found the "proportion weights" which play an important role in the steady state did not converge even though the training error did not change significantly in the training process. For system modeling, if the trained DNN is to replace the pump model, the DNN weights are set and hence no opportunity to change them exists during the testing (or in fact application) process.

In the next Chapter, some methods to overcome the low steady state accuracy problem are investigated.

CHAPTER 5 COMPENSATION FOR TRAINED DNN STEADY STATE

5.1 Introduction

The test results presented in Chapter Four indicated that the DNU based DNN was able to capture the dynamics of the "pump only" model in a certain frequency range (approximately less than 40Hz) but the steady state error was quite large. An examination on the last 5000 weight values showed that the "proportion weights" did not converge. In this Chapter, two compensation methods are presented which overcame the non-convergent "proportion weights" problem and eliminate the steady state error.

The compensation methods were formed by developing empirical functions to compensate for the changes in weights according to the relationships among four elements; that is, two system inputs, the trained DNN output and the error between the pump model output and the trained DNN output. In the first method, defined in this research as the "compensation equation method", the compensation function was developed by manually examining the relationships between these four elements point by point in the time response traces and then developing a set of empirical based equations. In the second method, as the static neural network is very effective in approximating either linear or nonlinear function, a static neural network was trained to learn a compensation input-output relationship in order to reduce the steady state error.

5.2 Compensation Equations Approach

5.2.1 Equations Development

The test results for the trained DNN showed that the steady state error between the "pump only" model and DNN model was quite large but the trained DNN appeared to capture satisfactorily the dynamics. One direct way to compensate for the steady state error was to quantify the error as a function of the magnitude of the inputs (essentially a three dimensional array of error vs the two inputs) and then to determine a compensating empirical equation forcing the output of the DNN to follow the "pump only" model output).

This approach was applied to the results shown in Chapter 4. Based on the observed relationship between the inputs and output error, compensation equations applied to the output of the DNN (*DNNoutput*) for this particular "pump only" model were obtained as:

$P_S = 1$ $\quad$ $\text{Output} = (0.3P_S + 0.9) * (DNNoutput) + \begin{cases} 0.1 & Pc > 0 \\ 0 & Pc \le 0 \end{cases}$ $\quad$ (5.1)

$P_S = 0.5$ $\quad$ $\text{Output} = (0.3Ps + 0.9) * (DNNoutput) + \begin{cases} 0.075 & Pc \ge 0 \\ 0 & Pc < 0 \end{cases}$ $\quad$ (5.2)

$P_S = 0$ $\quad$ $Output = (0.3Ps + 0.9) * (DNNoutput) + \begin{cases} 0.05 & Pc \ge 0 \\ 0 & Pc < 0 \end{cases}$ $\quad$ (5.3)

$P_S = -0.25$ $\quad$ $Output = (0.3Ps + 0.9) * (DNNoutput) + \begin{cases} 0 & Pc \ge 0 \\ 0.035 & Pc < 0 \end{cases}$ $\quad$ (5.4)

$P_S = -0.5$ $\quad$ $Output = (0.3Ps + 0.9) * (DNNoutput) + \begin{cases} 0 & Pc \ge 0 \\ 0.02 & Pc < 0 \end{cases}$ $\quad$ (5.5)

In the above equations, the term "*Output*" represents the output of the compensated DNN. The above equations were simplified to:

If Sign (P_S)*Sign (P_C) =1:

$$Output \approx (0.3Ps + 0.9) * (DNNoutput) + (0.05Ps + 0.05) \quad (5.6)$$

If Sign (P_S)*Sign (P_C) =-1:

$$Output \approx (0.3Ps + 0.9) * (DNNoutput) \quad (5.7)$$

Note that although the compensation equations were based on steady state data, the dynamic portions of the response were also amplified but, as will be shown, this did not have a significant effect overall.

5.2.2 Validity Testing

After applying these compensation equations to the trained DNN model, the compensated DNN model was subjected to series of tests which were similar to those in Chapter Four. As mentioned earlier, situations involving constant P_S with P_C varying or constant P_C with P_S varying would rarely occur in a load sensing pump; however, these extreme conditions illustrate how a pump model would respond to a single input. In the first series of tests, a sine wave was input into the model. Figure 5.1 shows the test results for P_S a constant, and P_C a sinusoidal signal. Figure 5.2 shows the test results for P_C a constant and P_S a sinusoidal signal. A test showing the case when both P_S and P_C were random sinusoidal signals is shown in Figure 5.3. P_S was then held constant and P_C varied as a random step input and the results shown in Figure 5.4. Results for P_C fixed as a constant and P_S a random step input are shown in Figure 5.5. In the last of this series of tests, both P_S and P_C were set to be random step inputs. The comparison of results are shown in Figure 5.6.

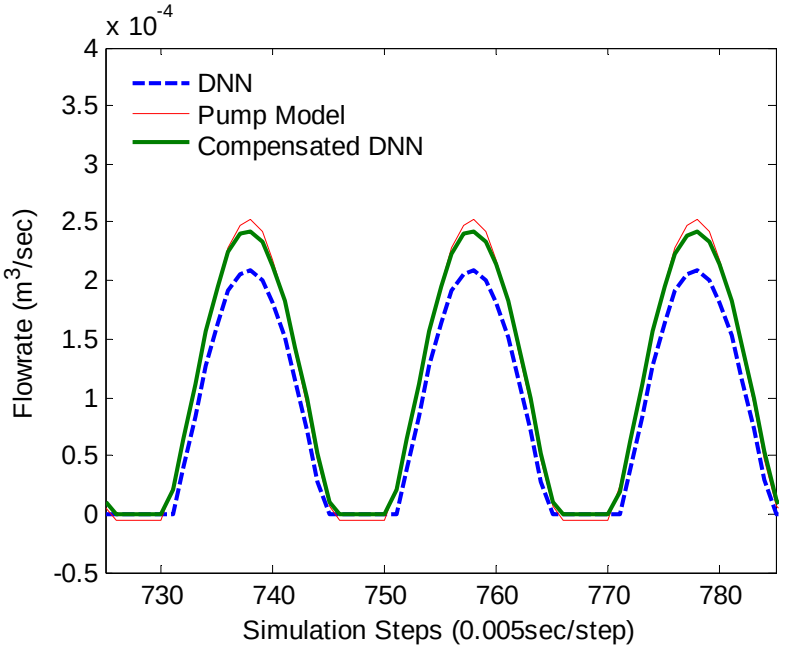


Figure 5.1 Comparison of the output flow of the trained DNN, the compensated DNN and the pump model (P_S constant and P_C varied in a sinusoidal fashion at 10Hz).

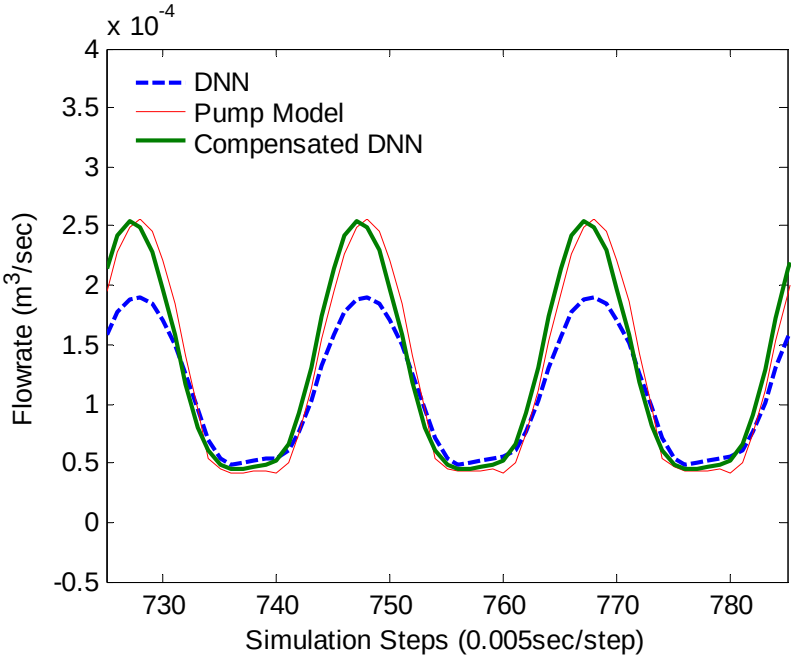


Figure 5.2 Comparison of the output flow of the trained DNN, the compensated DNN and the pump model (P_C constant and P_S varied in a sinusoidal fashion at 10Hz).

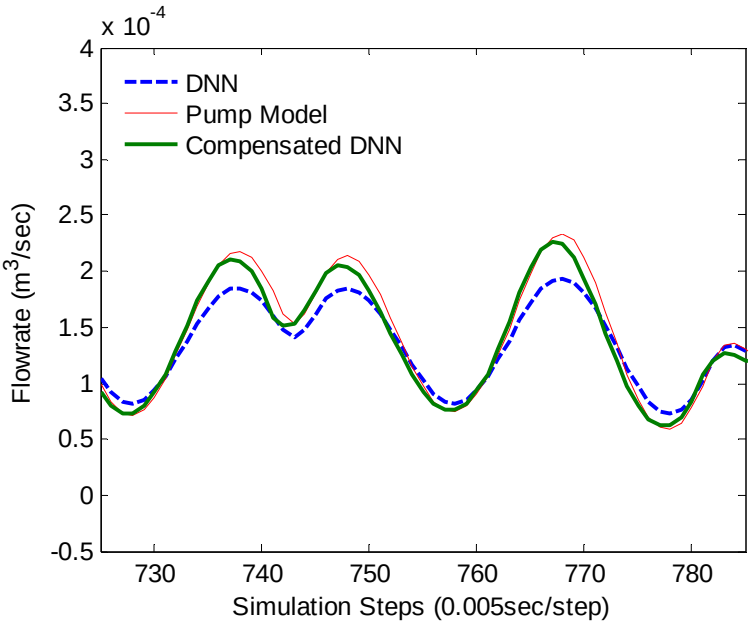


Figure 5.3 Comparison of the output flow of the trained DNN, the compensated DNN and the pump model (both P_S and P_C varied in a random sinusoidal fashion at 10Hz).

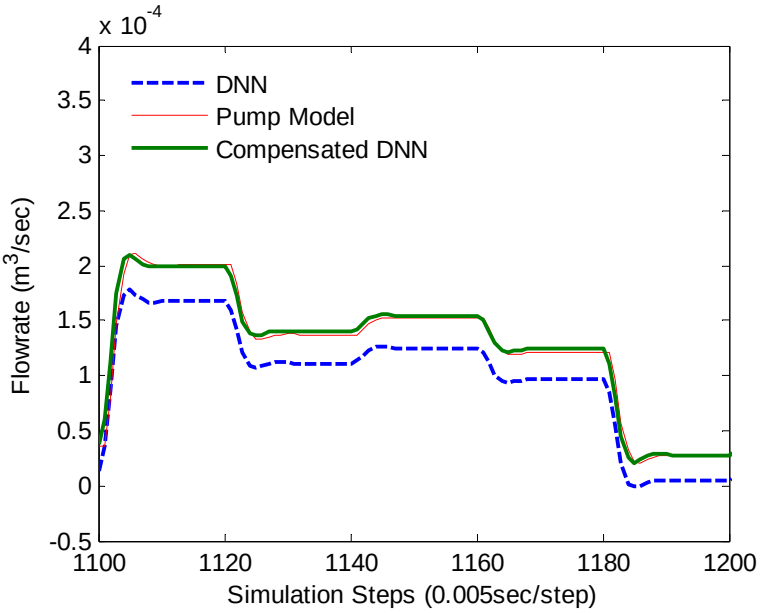


Figure 5.4 Comparison of the output flow of the trained DNN, the compensated DNN and the pump model (P_S constant and P_C varied in a random step fashion at 10Hz).

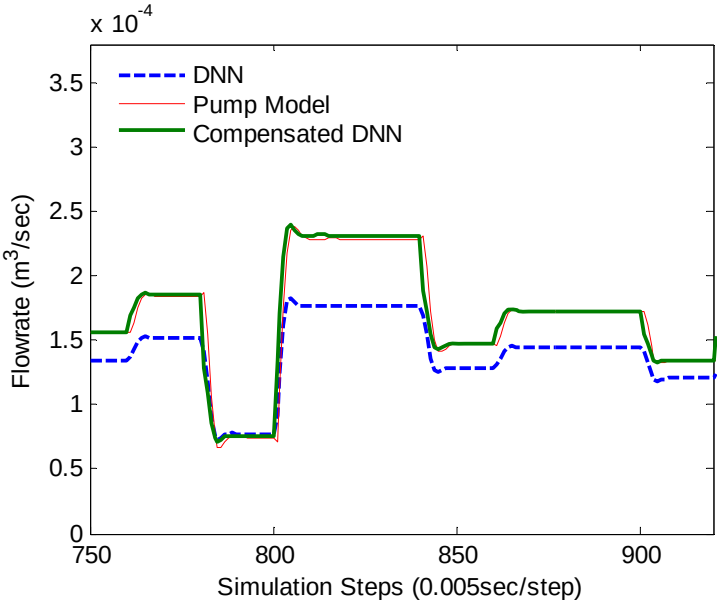


Figure 5.5 Comparison of the output flow of the trained DNN, the compensated DNN and the pump model (P_C constant and P_S varied in a random step fashion at 10Hz).

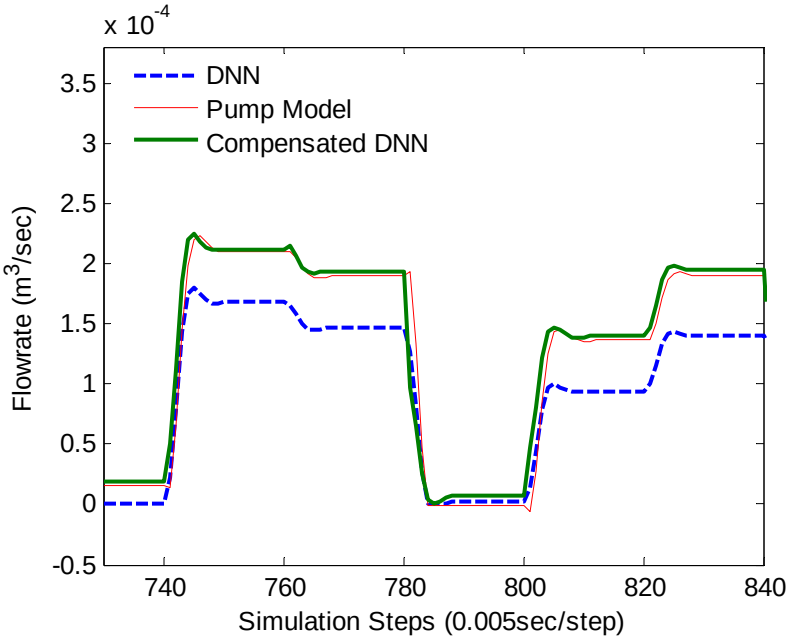


Figure 5.6 Comparison of the output flow of the trained DNN, the compensated DNN and the pump model (both P_S and P_C varied in a random step fashion at 10Hz).

The test results shown in Figures 5.1-5.6 indicate that both the steady state and the transient portions (step inputs) and amplitude (sinusoidal inputs) of the "pump only" and DNN models correlate quite well and hence the DNN can be adequately compensated with the proposed

scheme. However, as those compensation equations were developed manually, their validity cannot be guaranteed for all operating points and they were only effective for this particular pump model. The compensation equation approach was time consuming in its development, and demonstrated some accuracy limitation in practical application. Further, this "human interaction" seemed to reduce the effectiveness of the "black box" concept substantially. An alternate steady state compensation method which would provide higher accuracy and remove the human intervention aspect was needed.

5.3 Static Neural Network Compensation Approach

5.3.1 SNN Training Structure

The success in using compensation equations to eliminate the steady state error of the trained DNN demonstrated that the non-convergence "proportion weights" problem could be overcome by compensating the output of the DNN externally. It was well known that a static neural network (SNN) approach could approximate functions to any degree of accuracy as long as there were enough neurons in hidden layers(s) [Hornik, et al., 1989]. It was decided to train a SNN to replace the compensation equations in order to get higher accuracy over an extended application range.

A block diagram illustrating the steady sate SNN training process is shown in Figure 5.7. In this configuration, the two original system inputs, P_S and P_C, (which also were the inputs to the trained DNN and the "pump only" model), and the output of the trained DNN were fed into the SNN. The difference between the SNN output and the "pump only" model was used to train the SNN using an appropriate learning algorithm.

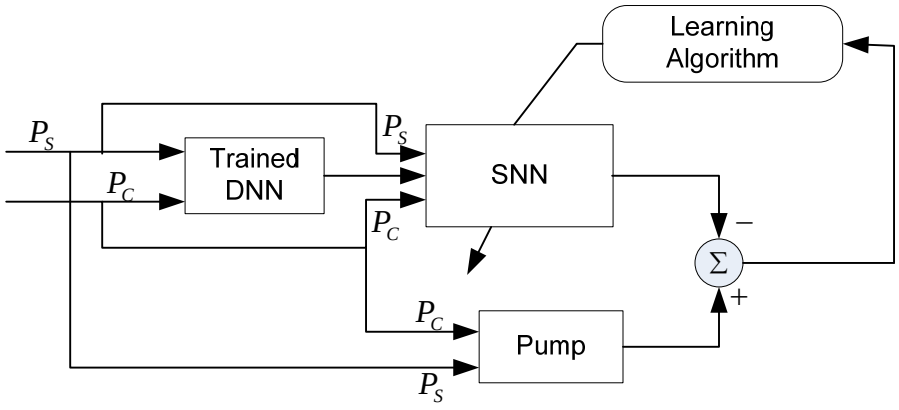


Figure 5.7 Block diagram for SNN training.

As mentioned above, a static neural network (SNN) is capable of reaching any degree of accuracy in function approximation as long as there are enough neurons in hidden layers(s); the more complex the SNN becomes, the more computation is required. After a suitable preliminary study, a neural model which included 10 neurons in the input layer, 4 neurons in the hidden layer and one neuron in the output layer was used for this study. The neural network is shown in Figure 5.8. Increasing the number of neurons above this produced negligible improvement in training accuracy but increased the required training time.

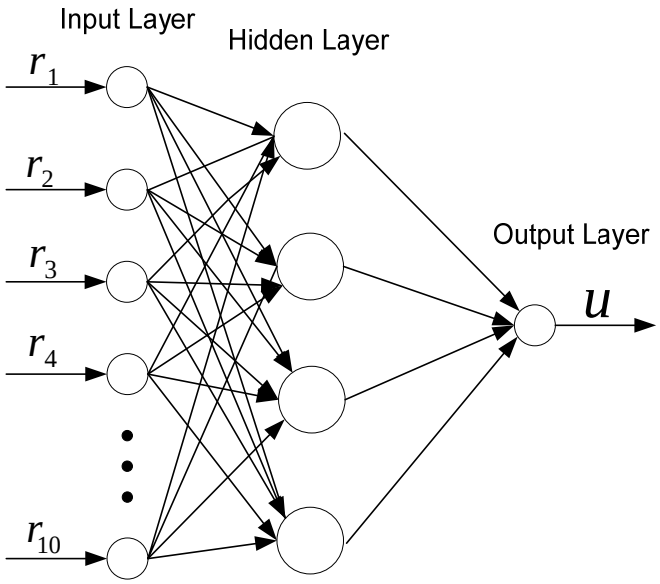


Figure 5.8 Schematic of SNN adopted for steady state compensation.

Since there was a readily available Static Neural Network Toolbox in Matlab, the steady state SNN training process was implemented using Matlab. There are several training algorithms available in the toolbox and a technique based on the Levenberg-Marquardt backpropagation technique was used in this study. The use of the Levenberg-Marquardt technique for neural networks was first described by [Hagan, et al., 1994]. This algorithm offers superior speed of convergence over simple backpropagation techniques and testing showed it performed better than the gradient descent techniques for the neural network model structure used.

As the SNN was used for the steady state compensation, the training signals, both P_S and P_C, were random step inputs with frequency of 10Hz. 2000 input pairs were collected to train the SNN using batch training (see Section 3.1). Both steady state and transient information were present in the epoch of training data.

5.3.2 Validity Testing for DNN &SNN

As the SNN training process approached completion, it was observed that the weights of the SNN converged to constant values (a required condition if the SNN was to be considered trained). The trained SNN (along with the previously trained DNN) was then tested using the structure shown in Figure 5.9.

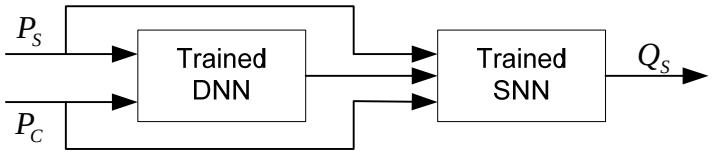


Figure 5.9 Structure of the trained DNN and SNN combination model.

A series of tests similar to those presented Section 5.2.2 were input into the combined neural network model. The test results for P_S $(12.9MPa)$ constant with sinusoidal P_C (10Hz) and then P_C $(3.4MPa)$ constant with sinusoidal P_S are shown in Figures 5.10 and 5.11. Figure 5.12

shows the test results when both P_S and P_C were varied in sinusoidal fashion with unrelated random amplitudes.

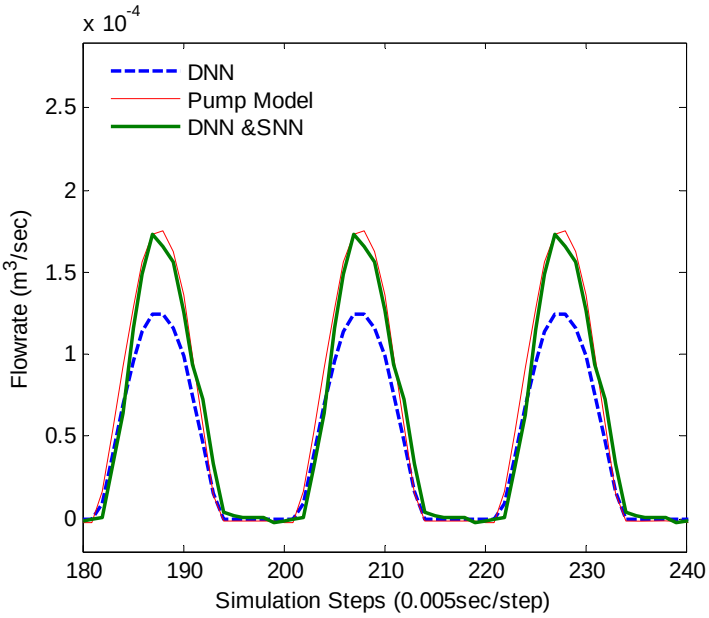


Figure 5.10 Comparison of the trained DNN, the DNN & SNN combination model and the pump model (P_S constant and P_C varied in a sinusoidal fashion at 10Hz).

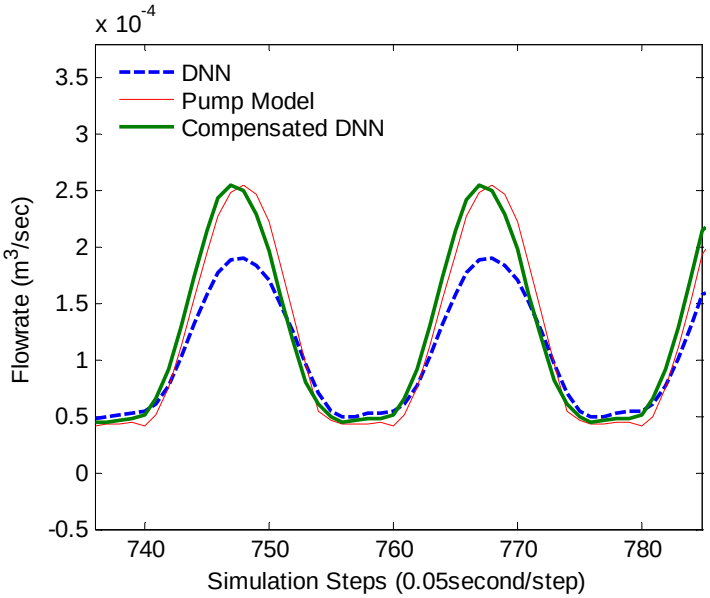


Figure 5.11 Comparison of the trained DNN, the DNN & SNN combination model and the pump model (P_C constant and P_S varied in a sinusoidal fashion at 10Hz).

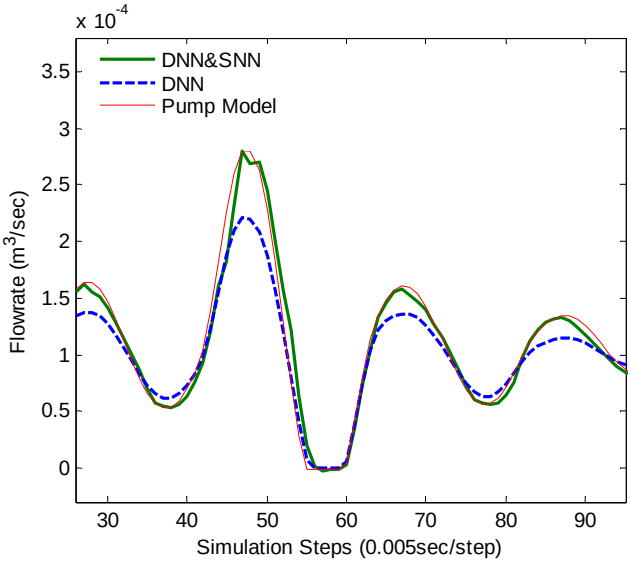


Figure 5.12 Comparison of the trained DNN, the DNN & SNN combination model and the pump model (both P_S and P_C varied in a random sinusoidal fashion at 10Hz).

A series of step input tests were also investigated. Figures 5.13, 5.14, and 5.15 show the test results where P_S was held constant with P_C varied in a random step fashion, P_C held constant with P_S varied in a random step fashion, and both P_S and P_C varied in a random step fashion respectively.

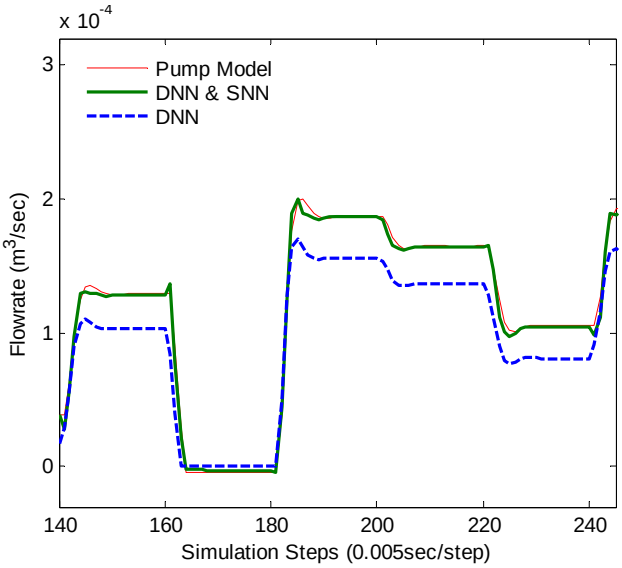


Figure 5.13 Step test results for DNN and SNN combination (P_S constant, P_C varied in a random fashion).

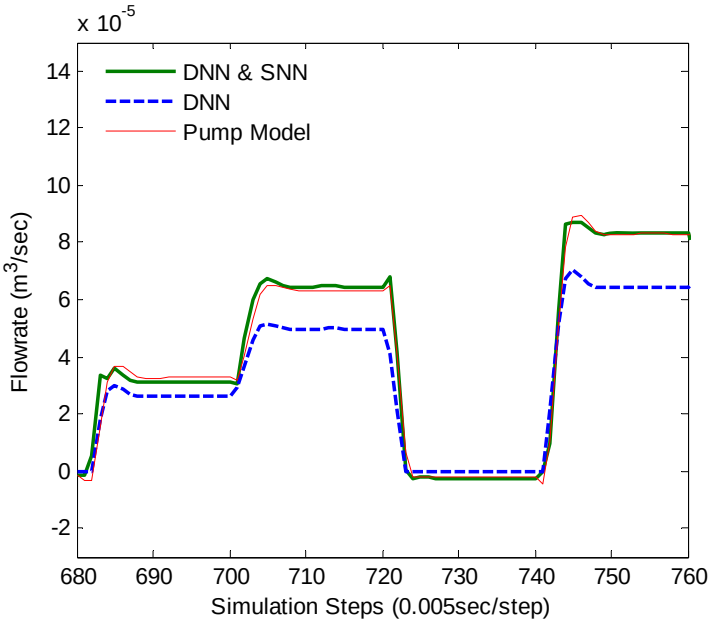


Figure 5.14 Step test results for DNN and SNN combination (P_C constant, P_S varied in a random fashion).

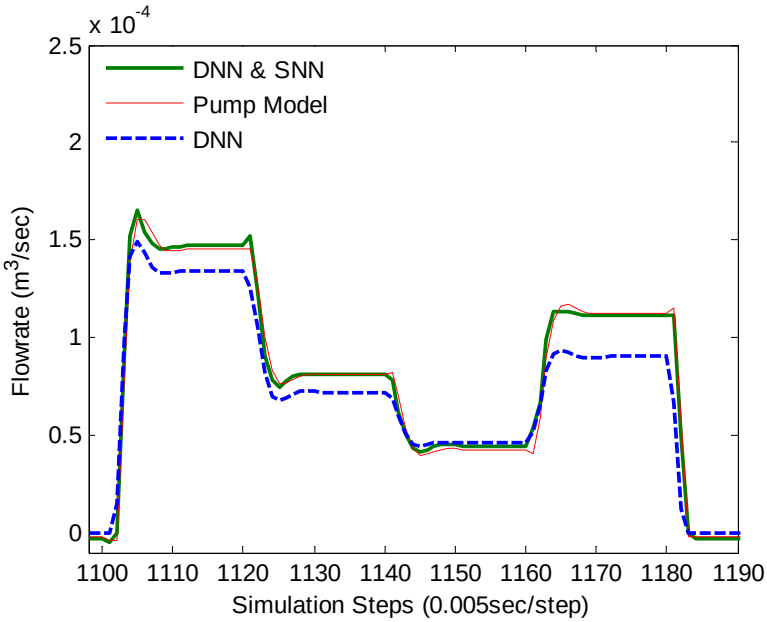


Figure 5.15 Step test results for DNN and SNN combination model (both P_S and P_C varied in a random fashion).

For completeness and comparison, the test results of a swept sinusoidal signal were also examined. The frequency response of the DNN and SNN combination and pump models where P_S was fixed at $12.93MPa$ and P_C amplitude varying from $2.8MPa$ to $3.4MPa$ is shown in Figure

5.16 and where P_C was held constant at $3.1MPa$ and P_S varied from $4.3MPa$ to $8.8MPa$ is shown in Figure 5.17.

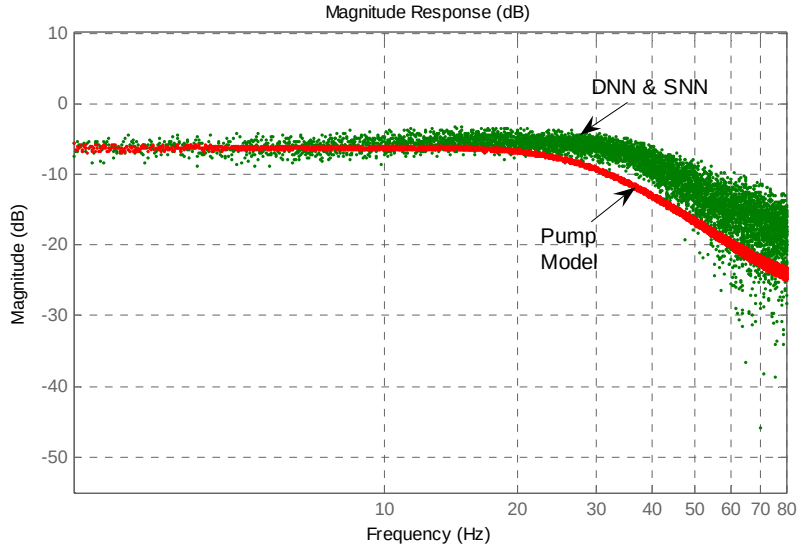


Figure 5.16 Magnitude frequency responses for DNN and SNN combination model ($P_S = 12.93MPa$. Magnitude $= 20\log\left(\frac{Q_S}{P_C}\right)$).

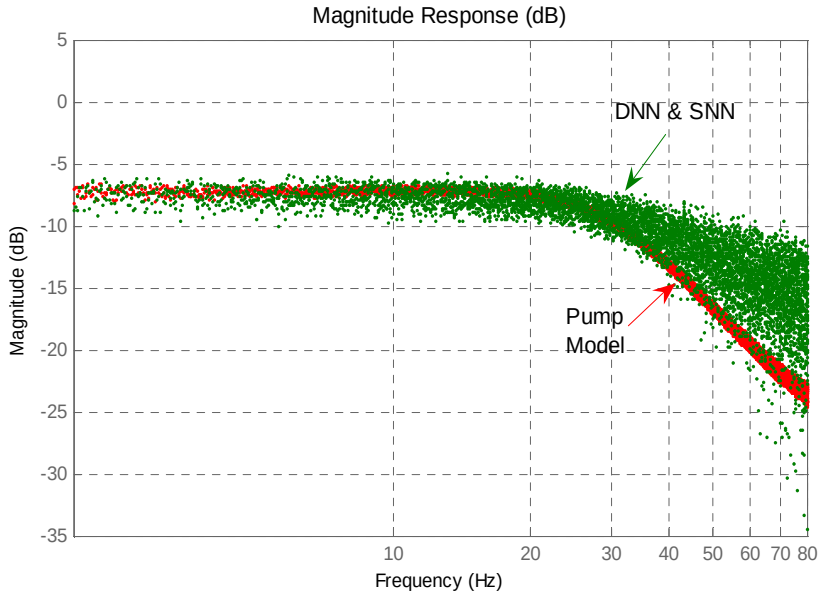


Figure 5.17 Magnitude frequency responses for DNN and SNN combination model ($P_C = 3.1MPa$. Magnitude $= 20\log\left(\frac{Q_S}{P_S}\right)$).

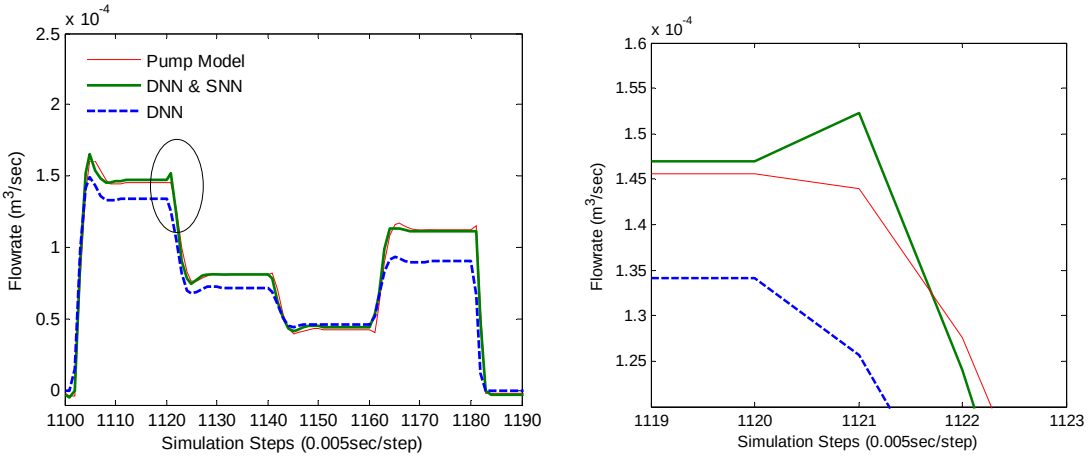
The results shown in Figures 5.10 – 5.15 do imply that the DNN and SNN combination model was able to represent both the steady state and the transient response of the "pump only" model. The application of SNN in eliminating the trained DNN steady state error was successful. The step and single sinusoid results are consistent with those illustrated in Figures 5.16 and 5.17 at lower frequencies (less than 35 Hz).

5.4 Discussion and Conclusion on Trained DNN Compensation Tests:

The test results presented in this Chapter shows that both the compensation equations and the SNN approach were effective in externally compensating the problem associated with the non-convergence of the "proportion weights" in the DNN. The steady state error of the trained DNN (which only captured the dynamics of the "pump only" model) was eliminated using two approaches. Compared to the compensation equation approach, the SNN approach was more straightforward and flexible and this approach was thought to have a wide range of application.

However, Figures 5.16 and 5.17 show that the magnitude frequency responses resolution of the DNN and SNN combination model becomes to be quite "blurry" at around 30Hz. This result was consistent with the observations and discussions in the Section 4.5.2 where it was observed that the accuracy of the trained DNN decreased above 40Hz due to the lower "richness" in higher frequency components (larger than 40 Hz). As a result, it was not possible for the output of the DNN and SNN combination be better than the trained DNN output since the trained DNN rather than the SNN determined the performance of the neural network model at higher frequencies. Furthermore, in order to modify the steady state, the SNN training signal frequency was fixed at 10Hz, and the SNN trained only at that frequency. Increasing the "richness" in frequency components for both DNN training and SNN training would be required to overcome this problem.

Occasionally, some spikes occurred in the DNN and SNN combination model output (Figure 5.18 (a) and (b)) but they are not universal. No adequate explanation for these spikes was found,



(a) (b)

Figure 5.18 “Spike” in the output of the DNN and SNN combination model.

In addition to the structure shown in Figure 5.7, two other “unsuccessful” training strategies were examined in this work. These are now briefly discussed.

5.4.1 SNN and DNN were trained simultaneously

A block diagram which illustrates how the SNN and DNN were trained simultaneously is shown in Figure 5.21.

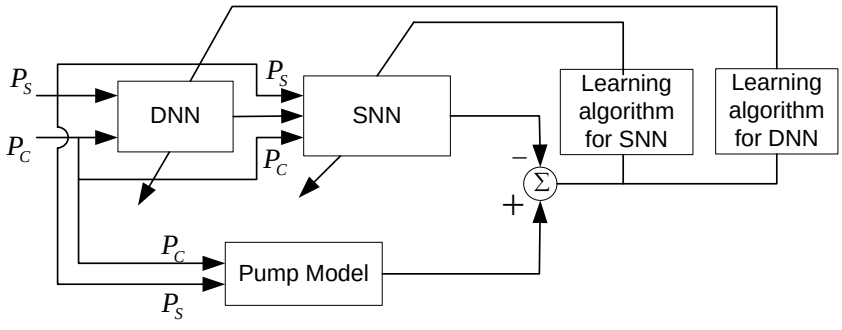


Figure 5.19 Structure for SNN and DNN trained simultaneously.

In the structure shown in Figure 5.19, both the steady state and the dynamic information were input to the SNN and DNN models and these were trained simultaneously using instantaneous training. During the training process, the SNN was exposed to both the steady state, and dynamic information, so it struggled to adjust the output to follow both which it could not do. Meanwhile, the DNN tried to learn both the steady state and dynamics which theoretically it should be able to do. As such, the SNN and DNN were in conflict in trying to learn the dynamics; that is the SNN interfered with the DNN in capturing the system dynamics. As a result, neither worked well and convergence of the error was not possible.

5.4.2 Only the steady state of the trained DNN was used to train the SNN

In this configuration, only the steady state information from the trained DNN output was used as an input to the SNN (shown in Figure 5.20). However, the test results showed this approach was unsuccessful in overcoming the non-convergence “proportion weights” problem.

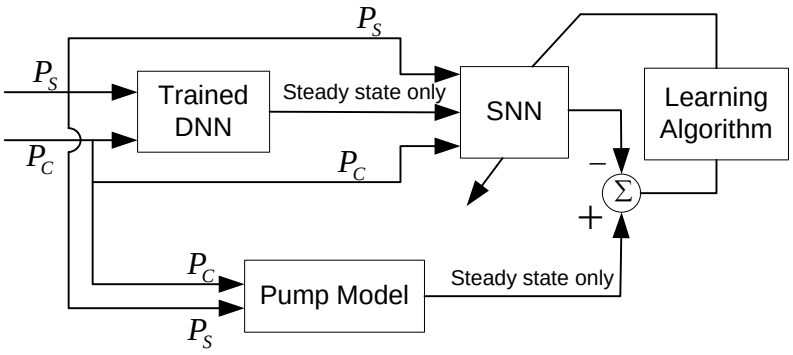


Figure 5.20 Structure for SNN trained by the steady state of trained DNN.

Considering the “pump model output” and “DNN output” shown in Figures 5.13, 5.14 and 5.15, the transient response and steady state of the trained DNN output had almost the same amount of shift from the “pump only” model which meant that not only the steady state, but also the transient response needed to be compensated. In the SNN training structure shown in Figure 5.20, the transient portion of the step response of the trained DNN was not used in the training

data for the SNN. Thus, the trained SNN would view the dynamic portion of the step response as steady state information and adjust it accordingly. This did distort the magnitude during the transient portion significantly and hence this approach was abandoned.

The final approach illustrated in Figure 5.7 overcame all the problems experienced in the structures shown in Figures 5.19 and 5.20; indeed, this final structure was powerful in eliminating the steady state error caused by the non-convergence "proportion weights" problem.

It should be pointed out that both the compensation equations and the SNN approach were used to eliminate the steady state error. The lower accuracy at higher frequency which caused by the lack of frequency components in the training epoch (see Section 4.5.2) can be overcome neither by compensation equation nor the DNN and SNN combination approach.

5.5 Summary

In this Chapter, two techniques using compensation equations and a static neural network were proposed to eliminate the steady state error caused by the non-convergence "proportion weights" problem in the trained DNN. In the compensation equations technique, the compensation equations were developed manually based on the observed relationships between P_S, P_L, the trained DNN output and the pump model output. This method was time consuming and had some limitations in the terms of the physical application (that is one group of equations were only effective over a certain operating range of a particular model). Because it was a straightforward approach and had a wide application range, the static neural network approach was adopted to compensate for the changes in the "proportion weights" of the trained DNN.

The test results for the compensation equations and the SNN approach indicated that these two techniques did increase the steady state accuracy; in addition, both had the capability to overcome the non-convergence “proportion weights” problem in the trained DNN.

CHAPTER 6 SUMMARY, CONCLUSIONS AND FUTURE WORK

6.1 Summary

The overall objective of this reach was to investigate the feasibility of using a dynamic neural unit (DNU) based dynamic neural network (DNN) in modeling a hydraulic component (specifically a load-sensing pump), which could be used in a simulation with any other required component models to aid in hydraulic systems design. To be representative of the component, the model must be valid for both the steady state and the transient response.

In Chapter two, three different pump model configurations ("pump, compensator, and valve" model, "pump and compensator" model and "pump only" model) were investigated. It was demonstrated that the "pump, compensator and valve" model was valid only for the particular control valve chosen; further, the applicability of the "pump and compensator" model was limited by "unique relationships", "dependent inputs" and "experimental data collection" problems. The "pump only" model avoided many of the issues related to other two models and was adopted in this study.

In Chapter Three, the Dynamic Neural Unit (DNU) which contained internal dynamic components (both feedforward and feedback time delays) was introduced as a means of capturing the dynamics of the pump. Several morphologies of the DNU were presented. A particular form of the DNU called DNU-3 was adopted and the gradient descent algorithm used to train the network was presented. Several multi-layer structures based on the DNN were also introduced. Based on previous knowledge of the order of the pump only model, a DNN structure comprising of two parallel DNUs was adopted for modeling purposes.

In Chapter Four, a mathematical model of a variable displacement pump (commonly found in a load sensing system) was developed using the Power Bond Graph technique. In order to achieve “richness” in both frequency and amplitude of the pump training signal (and subsequently, to the DNN), a special training signal (epoch) comprising of several different frequencies, was created in a Matlab environment. Special “testing” epochs were collected to validate that the trained neural network model could accurately reproduce the output response of the pump using input signals for which the DNN model was not trained. The results indicated that the trained DNN did capture the nonlinear dynamics of the pump model, but the steady state response was quite different. It was shown that a non-convergence of “proportion weights” was responsible for a large error in the steady state results.

In order to overcome some of the problems identified in Chapter Four, two techniques to eliminate the steady state error were proposed and implemented in Chapter Five. It was shown that both the compensation equations and the SNN approach were effective in improving the steady state performance of the trained DNN. Compared to the compensation equations, however, the SNN approach was superior due to its straightforward properties and because the concept has the potential to be used in other applications.

In summary, the research presented in this thesis examined the feasibility of using a DNU based DNN for simulating a dynamic model of a variable displacement pump (a “pump only” model). It was established that a DNU based DNN did capture the dynamics of the pump model. However, the accuracy of the steady state was poor but the compensation equations and the SNN approach did improve significantly the steady state performance.

6.2 Conclusions

Before listing conclusions, it is worthwhile to revisit the research objectives as defined in Chapter 1. These were:

- To find a more applicable model structure (that is, what are suitable inputs and outputs) for a load-sensing pump which will allow the model to be independent of the operation of the load sensing orifice and controlling positions.
- To apply DNU methodology to the neural network based pump model and to determine the new structure's ability to eliminate the error accumulation problems experienced in previous research.
- To develop, train and test an ANN based load-sensing pump model which can represent both the steady state and transient response of the pump over an expected operating range.

The first objective has been met in that a two input "pump only" model has been defined which overcame some of the problems associated with non-unique relationships between inputs and outputs. It was concluded that the two input "pump only" model (inputs compensator pressure P_C and pump pressure P_S and output flow Q_S) was the most appropriate form for simulation a load sensing pump using a black box approach. The "pump only" model morphology facilitated combining this model with other hydraulic components (control valve, load and compensator) in forming a system model of a load sensing unit.

The second objective (applying DNU methodology to the neural network based pump model to determine the new structure's ability to eliminate the error accumulation problems experienced in previous research) has also been accomplished by using two DNUs in parallel to form a DNN. Because this network had no external feedback connections, no error accumulation

was observed, an important benefit of the methodology. In the previous studies, the error accumulation problem limited the application of using neural networks in modeling the dynamic characteristics of a pump. It was concluded therefore that the DNN using two parallel DNU's could eliminate the error accumulation problem experienced in earlier studies.

The third objective stated above has also been met in that the DNN (compensated) was successfully trained to capture the dynamic and steady state characteristics of a load sensing pump. The network was trained using a dynamic backpropagation method based least mean-square algorithm and the network tested using a series of different input signal epochs that were not used in the training process. The DNN when used on its own did capture the dynamic characteristics of the pump but displayed a very large steady state error. The reason for the steady state error was also investigated and it was concluded that the non-convergent "proportion weights" of the DNN resulted in the steady state error.

To overcome this problem, two techniques, one consisting of manually derived compensation equations and one using a trained SNN placed after the DNN were studied. The test results demonstrated that both the "trained DNN with compensation equations" and the "trained DNN with the trained SNN" methodologies were able to represent the steady state over a specified operating range. The combined DNN and SNN was considered to be more appropriate over the trained DNN and compensating equation approach due to non-human intervention aspect of the approach. Therefore, it was concluded that it was feasible to use a DNN and SNN combination to represent (simulate) the static and dynamic characteristics of a "pump only" model over an acceptable operating range.

6.3 Future Work

It is believed that this study has generated a lot of challenges that need to be examined in future studies. These are:

a) Further investigation on the DNU structure and the DNN configuration is needed to address the non-convergence "proportion weights" problem in the DNN. In this study, the last 5000 values of the proportion weights were examined and they did not converge. This indicated that the DNN output was being driven by the error between the "pump only" model and the output of the DNN much in the same manner a DNN would be in a control application (rather than actually being trained). The reason for the non-convergence of the proportion weights needs to be investigated in depth as it has many implications for future applications of DNNs.

b) In the DNN morphology adopted in this study, all the weighted inputs were summed resulting in the same input going to the two DNUs (see Figure 3.17). As a consequence, the two DNUs were unable to differentiate between the two system inputs. Some modification to the DNN structure is needed such that the two DNUs can differentiate between the system inputs in an independent fashion. This would enable the DNN to see more information about the characteristics of the reference system and subsequently, improve the training accuracy. One possible DNN configuration that is forwarded is shown in Figure 8.1.

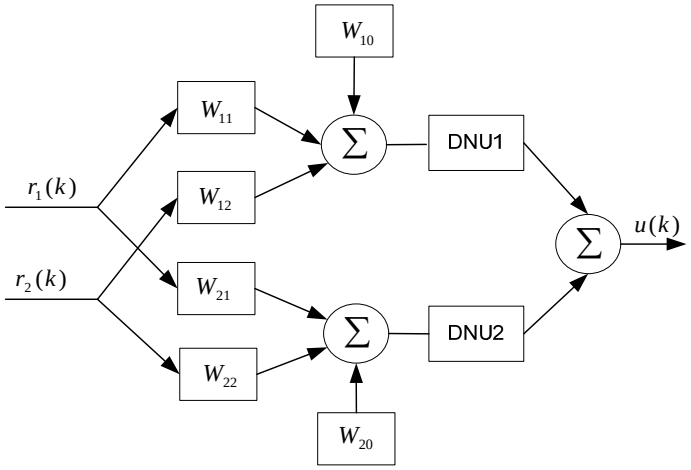


Figure 6.1 Recommended DNN structure for the future work

c) As a third recommendation, batch training as opposed to instantaneous training should be investigated in the DNN training process. In instantaneous training, the weights of the network are updated each time an input is presented to the network. Consequently, the DNN can "forget" part of the previous information. In batch training, the weights are only updated after all of the inputs in the epoch have been presented; the error function used for weights updating is the summation of the squared error of every step in epoch. However, due to the feedforward and feedback time delays in the DNU, employing batch training of DNU could be a challenge.

d) It is also recommended that further investigation of the combination of the SNN and DNN be conducted. It has been shown that the DNUs based DNN was able to represent the dynamics of a nonlinear system such as a pump, and that the SNN was very powerful in reproducing the steady state. In this thesis, only a very preliminary fundamental exploration on the SNN and DNN combination was conducted. Because of its success, it is believed that further development of this approach is warranted.

e) Since comparison of the system response of the reference model and the trained neural network model over certain frequency range played an important role in evaluating the performance of the neural network model, it is suggested that a frequency response method valid

for a two input nonlinear system be pursued. Unlike a single input system, testing the system response of a nonlinear system over a large frequency range is very hard for the two- input system. A verified method for the frequency response of multi-input nonlinear systems would be very valuable in future studies.

f) It was demonstrated that the neural network model performance deteriorated at frequencies higher than 40Hz. It is recommended that the epoch training signal include more high frequency components in both the DNN training and SNN training signals to improve the richness of the training epoch.

g) As a final recommendation, all the models presented in this thesis should be applied to a “real” system, in order to establish the feasibility of using the DNU based DNN in modeling a physical component such as pump found in a load-sensing hydraulic system.

LIST OF REFERENCES

1 Anderson, J.A., 1983. Cognitive and Psychological Computation with Neural Models. *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 13, pp. 799-815.

2 Astrom, K. J. and Eykhoff, P., 1997. System Identification- A Survey. *Automatica*, Vol. 7, pp. 123-162.

3 Bailey, S., Watton, J., 2002. A Neural Network Approach to Transmission Line Modelling and Fault Diagnosis of a Hydraulic Press Control System. *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, Vol. 216, No. 4, pp. 357-367.

4 Bellman, R. and Astrom, K. J., 1970. On structural Identifiability. *Mathematical Biosciences 7*, pp. 329-340.

5 Bitner, D. and Burton, R.T. 1984. Experimental Measurement of Load-Sensing Pump Parameters. *Proceeding of the 40th National Conference on Fluid Power*, Chicago, pp.153.

6 Book, R. and Goering, C. E. 1997. Load sensing Hydraulic System Simulation. *Applied Engineering in Agriculture.* ASME Vol1. 13(1), pp.17-25.

7 Chen, S., Billings, S. A., 1992. Neural Networks for Dynamical System Modeling and Identification. *Int. J. Control*, Vol. 56, No. 2, pp. 319-346.

8 Chinniah, Y. A., Burton, R., Habibi, S., 2001. Parameter Estimation in a Hydrostatic System using Extended Kalman Filter. *American Society of Mechanical Engineers, The Fluid Power and Systems Technology Division (Publication) FPST*, Vol. 8, *Fluid Power Systems and Technology - 2001*, pp. 1-6.

9 Deshpande, N.A, and Gupta, M.M., 1998. Inverse kinematic neuro-control of robotic systems, *Engineering Applications of Artificial Intelligence*, Vol.11, No. 1, pp. 55-66.

10 Erkkila, M. 1999. Practical Modeling of Load Sensing Systems. *Proceedings of the Sixth Scandinavian International Conference on Fluid Power*, SICFP'99, Tampere, Finland, pp. 445.

11 Fukushima, K., Miyake, S., and Ito, T., 1983. Neocognitron: A neural Network Model for a Mechanism of Visual Pattern Recognition. *IEEE Tans. Systems, Man and Cybernetics*, Vol.13, No. 5, pp. 826-834.

12 Gupta M. and Rao, D., 1993. Dynamic neural Units with Applications to the Control of Unknown Nonlinear Systems. *The Journal of Intelligent and Fuzzy Systems*, Vol. 1, No. 1, pp. 73-92.

13 Gupta, M. and Rao, D., 1992. Dynamic Neural Units in the Control of Linear and Nonlinear Systems. *Int. Joint Conf. On Neural Networks (IJCNN)*, Baltimore, pp. 100-105.

14 Habibi, S.R., Burton, R., Parameter identification for a high performance hydrostatic actuation system using the variable structure filter concept. *American Society of Mechanical Engineers, The Fluid Power and Systems Technology Division (Publication) FPST*, v 11, *Proceedings of the ASME Fluid Power Systems and Technology Division - 2004*, pp. 93-101.

15 Hagan, M. T., and Menhaj, M. (1994). Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transaction on Neural Networks*, Vol. 5, N0. 6, pp. 989-993.

16 Harris, R., Edge, K., and Tilley, D., 1974. The Spin Motion of Pistons in a Swashplate Type Axial Piston Pump, *Proceeding of the Institution of Mechanical Engineers*, Vol. 188, No. 17.

17 Haykin, S. S., 1994. Neural networks: a comprehensive foundation. *Maxwell Macmillan International*, New York.

18 Hecht-Nielsen, R., 1988. Neurocomputing: Picking the Human Brain. *IEEE Spectrum*, Vol. 25, pp. 36-41.

19 Hindman, J. J., 2002. Condition Monitoring of Valves and Actuators in a Mobile Hydraulic system Using an Artificial Neural Network and Expert Data. *M.Sc. Thesis,* Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Saskatchewan.

20 Hopfield, J.J., 1984. Neurons with Graded Response Have Collective Computational Properties Like of Those Two-State Neurons. *Proc. Of the National Academy of Sciences*, Vol. 81, pp. 3088-3092.

21 Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer Feed Forward Networks are Universal Approximators. *Neural Networks*, 2(5), pp. 359-366.

22 Hunt, K.J., Sbarbaro, D., Zbikowski, R., and Gawthrop, P.J., 1992. Neural Networks for control Systems- A Survey. *Automatica*, Vol. 1, No. 6, pp. 1083-1112.

23 Hush, D.R., and Horne, B.G., 1993. Progress in Supervised Neural Networks. *IEEE Signal Processing Magazine,* Vol. 10, No. 1, pp. 8-39.

24 Ivantysyn, J. and Ivantysynova, M., 2000. Hydrostatic Pumps and Motors, Principles, Design, Performance, Modeling, Analysis, Control and Testing. *Academia Books International*, New Delhi (ISMB 81 -85522-16-2).

25 Ivantysynova, M. 2003. Prediction of Pump and Motor Performance by Computer Simulation, *First International Conference on Computational Methods in Fluid Power Simulation*, Melbourne, Australia, pp. 505-522.

26 Kavanagh, G., Schoenau, G., and Burton, R., 1990. Dynamic Analysis of a Variabel

Displacement Pump, *Transactions of the ASME Journal of Dynamic Systems and Control*, Vol. 112, No.1, pp. 122.

27 Kim, S. D., Cho, H. S. and Lee, C. O., 1987. A Parameter Sensitivity Analysis for the Dynamic Model of a Variable Displacement Axial Piston Pump. *Proceeding of Institute of Mechanical Engineers*, USA, Dec. 13-18.

28 Kim, S. D., Cho, H. S. and Lee, C. O., 1987. On the Order Reduction of the Dynamic Model of a Pressure- Compensated Variable Displacement Hydraulic Pump. *Presented at the ASME Winter Annual Meeting. Boston, Massachusetts*, USA, Dec. 13-18.

29 Kim, S., Aker, A. and Zeiger, D., 1989. Oil Entrapment in an Axial Piston Pump and its Effect upon Pressures and Swash Plate Torques, *Proceeding of the National Conference on Fluid Power and Control*, Chicago, pp.3.

30 Krus, P. 1988. On Load Sensing Fluid Power Systems. *Dissertation No. 198*. Linkoping University, Sweden.

31 Lamontagne, D., 2002. Investigations in Modeling a Load-Sensing Pump Using Neural Networks, M.Sc. Thesis, Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Saskatchewan.

32 Lamontagne, D., Burton, R., Ukrainetz, P., and Ruan, J., 2003. Investigations in Modeling a Load-Sensing Pump Using Neural Networks. *Proceeding of the Fourth International Symposium on Fluid Power Transmission and Control (ISFP'2003),* Wuhan, PRC, pp 151-158..

33 Lantto, B., Krus, P., and Palmberq, J., 1993. Dynamic Properties of Load-Sensing Systems with Interacting Complex Mechanical Loads, *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME,* Vol. 115, No.3, pp. 525-530.

34 Lantto, b., Palmberg, J., and Krus, P., 1990. Static and Dynamic Performance of Mobile Load-Sensing Systems with Two Different Types of Pressure-Compensated Valves. *SAE Transactions*, Vol. 99, N0. 1, pp. 251-265.

35 Manring, N.D., 2005. Hydraulic Control Systems. *John Wiley & Sons*. NY.

36 McNamara, J., Edge, K., and Vaughan, N., 1997, Hybrid Analytical/Neural Network Model of Variable Displacement Pump Dynamics, American Society of Mechanical Engineers, *The Fluid Power and Systems Technology Division (Publication) FPST, v 4, Fluid Power Systems and Technology*, pp. 71-76.

37 Moller. J.R,. 1990. Load-Sensing Pumps Have Their Time Come? *Hydraulic and Pneumatic Magazine*, pp. 57.

38 Narendra, K. S., and Parthasarathy, K., 1989. Neural Network and Dynamic Systems-Part II : Identification. *Report No. 8902*, Yale University.

39 Narendra, K. S., and Parthasarthy, K., 1990. Identification and Control of Dynamic Systems Using Neural Networks, *IEEE Trans. Neural Networks*, Vol. 1, No. 1, pp. 4-27.

40 Oysal, Y., 2005. A Comparative Study of Adaptive Load Frequency Controller Designs in a Power System with Dynamic Neural Network Models, *Energy Conversion and Management*, Vol. 46, No. 15-16, pp. 2656- 2668.

41 Poggio, T., and Girosi, F., 1990. Networks for Approximation and Learning. *Proc. IEEE*, Vol. 78, No. 9, pp. 1481-1497.

42 Richards, C.R., Tomlinson, S.P., Tilley, D.G., and Burrows, C.R., 1989. Bath fp-A Second Generation Simulation Package for Fluid Power Systems. *9th International Symposium on Fluid Power*, Cambridge, U.K.

43 Schoenau, G. J., Burton, R., Kavanagh, G. P., 1990. Dynamic analysis of a variable displacement pump. *Journal of Dynamic Systems, Measurement and Control, Transactions ASME*, Vol. 112, No. 1, pp. 122-132.

44 Soderstrom, T. and Stoica, P., 1989. System Identification. *Prentice Hall*, New York.

45 Song, T.M, 2001. Development of Dynamic Neural Networks Units with Control Applications, *M.Sc. Thesis,* University of Saskatchewan, Saskatoon.

46 Song, Y., and Gupta, M.M., 1999. Dynamic Neural Controller for Complex Nonlinear Systems, *IEEE North Saskatchewan Syposium*, Saskatoon, pp. 25-40.

47 Song, Yinmin, and Gupta, M.M., 1999. Dynamic Neural Controller for Complex Nonlinear Systems, *IEEE North Saskatchewan Syposium*, Saskatoom, pp. 25-40.

48 Srivastava, V., 1998. Neural-control of Unknown Dynamic Systems. *M.Sc. Thesis*, University of Saskatchewan, Saskatoon.

49 Wasserman, P. D., 1989. Neural Computing: Theory and Practice, *Van Nostrand*, New York.

50 Watton, J., and Xue, Y., 1997. Simulation of Fluid Power Circuits Using Artificial Network Models (Part 1 and 2), *Proc, Instn. Mech.engrs.- Part* I , Vol.211, pp.417-438.

51 Werbos, P. J., 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences.

52 Widrow, B., and Lear, M.A., 1990. 30 Years of Adaptive Neural Networks Perception, Madaline and Backpropagaton. *Proceeding of the IEEE*, Vol. 8, No. 9.

53 Wright, J, A., 1988. The Design of a Microcomputer Control System for a Variable Displacement Hydraulic Pump. *M.Sc. Thesis,* Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Saskatchewan.

54 Wright, J. Burton R., and Schoenau, G.., 1992. Design of a Microcomputer Control System for a Variable Displacement Hydraulic Pump, *The Journal of Fluid Control*, Vol. 21, No. 2-3, pp. 116-140.

55 Wu, D., 2002. Steady State Analysis of the Load Sensing Systems, *Proceeding of the Second International Ph.D. Conference*. FPNI, Modena, Italy, July.

56 Wu, D., Burton, R. Schoenau, G., and Bitner., D., 2002. Establishing Operating Points of a Linearized Model of a Load Sensing System, *International Journal of Fluid Power,* Vol. 3, No. 2. pp. 47-54.

57 Wu, D., Schoenau, G., Burton, R., Bitner, D., 2005. Model and Experimental Validation of a Load Sensing System with a Critically Lapped Regulator Spool. *International Journal of Fluid Power,* Vol. 6, No. 3, November, 2005, pp. 5-18.

58 Xu, X. P., 1997. Experimental Modeling of a Hydraulic Load Sensing Pump Using Neural Networks. *Ph.D. Thesis*, Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Saskatchewan.

59 Xu, X. P., Burton, R. and Sartent, C., 1997. Experimental Implementation of a Neural Simulator. *Fluid Power Systems and Technology*, Vol. 4, November, pp.21- 26.

60 Xu, X.P., Sargent, C. and Burton, R., 1996. Experimental Identification of a Flow Orifice Using a Neural Network and the Conjugate Gradient Method. *Transactions of ASME: Journal of Dynamic Systems, Measurement, and Control*, Vol. 118, June, pp. 273-277.

61 Zeiger, G., Akers, A., 1986. Dynamic Analysis of an Axial Piston Pump Swashplate Control. *Proceedings of the Institution of Mechanical Engineers, Part C: Mechanical Engineering Science*, Vol. 200, No. C1, pp. 49-58.

APPENDIX A The Concept of Power Bond Graph

A.1 Introduction

Bond graph is a dual-signal flow diagram comprising of numbers of terms and is a powerful graphical/analytical tool for capturing the energy structure of systems. From the Bond Graph, a series of dynamic equations which reflect proper causality and hence are “computer ready” can be developed. Power Bond Graph increases the insight into systems behaviour. In this thesis the Power Bond Graph technique is applied to a hydraulic control system which consists of a compensator, variable swashplate pump, load sensing valve and load. An excellent description the Power Bond Graph as applied to a hydraulic system can be found in [Dransfield] and indeed, this introduction is a summary of sections of this text.

A.2 Bond Graph Terms and Symbols

Terms and symbols are major components in Power Bond Graph. Based on these terms and symbols and their appropriate arrangement, a series of expression equations can be developed which reflect proper causality and which define the dynamic behaviour of the system.

Effort and Flow Variables

The term effort and flow are used to represent the potential and the flowrate respectively. For example, pressure and force are efforts, and volumetric flowrate and velocity are flow variable. Generally, E and Q are used as the symbols for effort and flow respectively. However, other symbols can be and are used.

Sources

The constant power variable is called a source. The effort source is represented by SE and the flow source is denoted by SQ.

Power Bonds

A power bond is a line standing for the route of the power flow and is equipped with the related effort and flow variables. The typical power bond is:

$$\frac{E}{Q}$$

Power Transformers

The symbol TF is adopted to represent the power transformation generated by some hydraulic devices, such as electric motor, hydraulic pump and others. For example, the symbol

$$\frac{E1}{Q1} \quad \mathsf{TF} \quad \frac{E2}{Q2}$$

means the power $E1 \cdot Q1$ converts to the power $E2 \cdot Q2$ and $E1 \cdot Q1 = E2 \cdot Q2$.

Dynamic Effects

Three elements contribute to the dynamics of a control system, and they are referred to as resistive power dissipation, R, capacitive power storage, C, and inertive (or inductive) power storage, I. In a hydraulic control system, a resistive effect includes all forms of the friction and pressure drops in the pipe line. Capacitive power storage is generated from devices which can store and release potential energy to affect the dynamics of the system, such as mechanical springs, and hydraulic accumulators. Inertive power storage is associated with the acceleration and deceleration of the inertias(fluid and mechanical).

Summing Junctions

There are two different summing junctions which are referred as a0-junction and 1-junction. They are denoted by:

The 0-juction means

$$Q1 + Q2 + Q3 = 0$$

$E =$ same in all three bonds.

The 1-junctin means

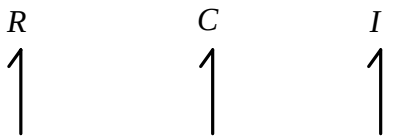
$$E1 + E2 + E3 = 0$$

$Q =$ same in all three bonds.

Power Flow Directions Causality

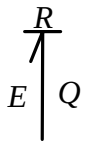
Direction of power flow

In the power bond graph, arrowheads are adopted to indicate the power flow direction which actually is from the source to the load. Specially, a half arrowhead is adopted to represent the R, C and I element as follows:



Causality

A short transverse bar, called as causal bar, is adopted to indicate the cause variable and the effect variable in a power bond. The arrangement of the causal bar is very important, because it determines the cause-effect relationship between the two variables in one power state, and therefore determines the form of the equations. For example,



indicates that an R expression equation should be

$$Q = R(E)$$

In this situation, E is the cause variable and Q is the effect variable. Usually, an I element receives effort (cause) and generates flow (effect), and a C element receives flow and generates effort. Since the resistive or dissipative elements do not have time integral form, an R element can have any type of causal structure.

At a 1-junction, only one bond should bring the information of flow; i.e., only one bond should be open end and all others should be stroked. Similarly, at a 0- junction, only one bond should be stroked nearer to the junction.

As a final step, a series of equations which are readily for computer simulation are developed based on the power bond graph technique.

A.3 Example

In order to demonstrate the procedure of the bond graph development, a simple example is presented. Figure A.1.1 illustrates the schematic of a simple hydraulic system. Figure A.1.2 shows the power bond graph structure for the system, and the equations which describe the dynamic characteristics of the hydraulic system are also forward.

Each power bond in the horizontal line of Figure A.1.2 represents a physical component through which power flows to the next component. The product of the two variables associated with the power bond describes the power flowing. Thus, $P_S \cdot Q_S$ (supply pressure times supply

flowrate) describe the power flowing into the control valve, and $F_a \cdot \dot{X}_m$ (force applied to the mass times velocity of the mass) describes the power flowing into the load. The R, C and I terms illustrate that the resistive (R), capacitive (C), and inertive (I) terms that affect the system's dynamic response. The causality which is indicated by a short transverse bar at one end of each line shows a cause-effect determination.

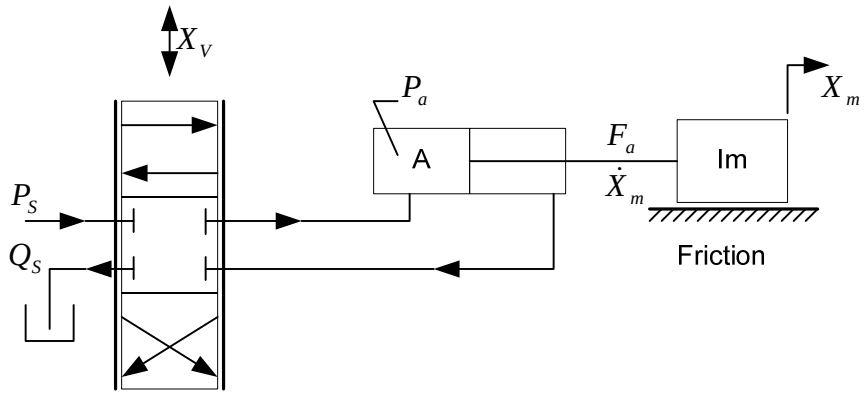


Figure A.1.1 Schematic of a hydraulic system.

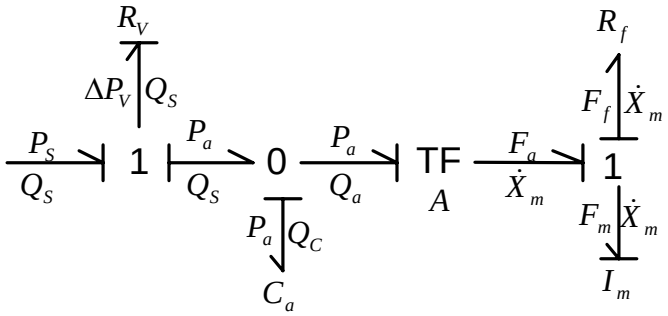


Figure A.1.2 Power bond graph structure of the hydraulic system.

Equations

For this Bond Graph, the dynamic equations are:

$P_S =$ constant (assumption), (A.1.1)

$Q_S = K \cdot X_V \cdot \sqrt{\Delta P_V}$ (for X_V positive), (A.1.2)

$P_a = P_a(0) + 1/C_a \cdot \int Q_C dt$ (linear capacitance equation), (A.1.3)

$F_f = R(X_a) =$ friction force for cylinder, (A.1.4)

$F_m = F_a - F_f$, (A.1.5)

$\dot{X}_m = \dot{X}(0) + {}^{1}\!/\!_{I_m} \cdot \int F_m dt$ (Newton's law), (A.1.6)

$F_a = P_a \cdot A$, (A.1.7)

$Q_a = {}^{\dot{X}_m}\!/\!_{A}$, (A.1.8)

$\Delta P_V = P_V - P_a$, (A.1.9)

$Q_C = Q_S - Q_a$. (A.1.10)

Observation of these equations reveals that the unknown states appear only once on the left hand side and hence causality problems in computer implementation are avoid.

APPENDIX B Pump Simulation Model Parameters

Table B.1 is a listing of the values of all parameters and coefficients of the pump model adopted as a reference model in this study. These values were obtained from [Wright, Kavanagh].

Table B.1: The values of the pump model parameters

Parameters	Description	Value	units
S_{ω}	Pump rotation speed	183.3	rads/sec
R_l	Leakage resistance coefficient of pump	361.8×10^{-15}	$m^3/(p_a\cdot\sec)$
A_{cp}	Area of control piston face	239.4×10^{-6}	m^2
k_{spr}	Angular spring constant of return spring	56	$N\cdot m/rads$
R_{yd}	Lumped damping resistance of swash plate assembly	0.422	$N\cdot\sec$
I_{yk}	Effective mass moment of inertia of yoke	1.32×10^{-3}	$kg\cdot m^2$
$T_{spr}(0)$	Pretension of return spring	18.5	$N\cdot m$
k_{pr1}	Coefficient of load torque #1	0.128	$N\cdot m$
k_{pr2}	Coefficient of load torque #2	0.725×10^{-6}	$N\cdot m/P_a$
k_{pr3}	Coefficient of load torque #3	0.625×10^{-6}	$N\cdot m/(P_a\cdot rads)$
k_{pr4}	Coefficient of load torque #4	-0.0962	$N\cdot m\cdot\sec$

APPENDIX C Derivation of the "Pump Only" Model Transfer Function

Initial feasibility studies on the selection of the DNN's structure, it was deemed important to understand what the order of the pump was. This Appendix then linearizes the nonlinear equations of the pump and develops a dual input transfer function.

In Chapter Four, the describing equations that were generated from the Power Bond Graph of the "Pump Only" model are repeated as:

$$Q_p = (S_\omega)(k_\omega).\theta\,, \tag{B.1}$$

$$Q_l = R_l \cdot P_s\,, \tag{B.2}$$

$$Q_s = Q_p - Q_l\,, \tag{B.3}$$

$$T_{yk} = A_{cp} \cdot b \cdot P_C\,, \tag{B.4}$$

$$T_{spr} = k_{spr}\int_0^t \dot{\theta} dt + T_{spr}(0)\,, \tag{B.5}$$

$$T_{lp} = k_{pr1} + k_{pr2}P_s + k_{pr3}P_s\theta + k_{pr4}\dot{\theta}\,, \tag{B.6}$$

$$T_{yd} = R_{yd}\dot{\theta}\,, \tag{B.7}$$

$$T_{net} = T_{yk} - T_{spr} - T_{yd} - T_{lp}\,, \tag{B.8}$$

$$\dot{\theta} = \frac{1}{I_{yk}}\int_0^t T_{net} dt + \dot{\theta}(0)\,. \tag{B.9}$$

In order to simplify the derivation, some of the coefficients have been simplified as:

$$Q_p = K_1\theta\,, \tag{B.10}$$

$$Q_l = K_2 \cdot P_s\,, \tag{B.11}$$

$$T_{spr} = K_3\theta + K_4\,, \tag{B.12}$$

$$T_{lp} = k_5 + k_6 P_s + k_7 P_s\theta + k_8\dot{\theta}\,, \tag{B.13}$$

$$T_{yd} = K_9\dot{\theta}\,, \tag{B.14}$$

$$T_{yk} = K_{10}\cdot b\,, \tag{B.15}$$

$$\dot{\theta} = \frac{1}{K_{11}}\int_0^t T_{net}\,dt + \dot{\theta}(0)\,. \tag{B.16}$$

where $K_1 = S_\omega \cdot K_\omega$, $K_2 = R_l$, $K_3 = K_{spr}$, $K_4 = T_{spr}(0)$, $K_5 = K_{pr1}$, $K_6 = K_{pr2}$, $K_7 = K_{pr3}$, $K_8 = K_{pr4}$, $K_9 = R_{yd}$, $K_{10} = A_{cp}\cdot b$, $K_{11} = I_{yk}$.

There is a nonlinear component, $K_7 \cdot P_S \cdot \theta$, in Equation (B.13) and it need to be linearized. This is done using a Taylor's series expansion about a particular operating point as:

$$F = K_7 \cdot P_S \cdot \theta = F_1 + \left.\frac{\partial F}{\partial P_S}\right|_1 \Delta P_S + \left.\frac{\partial F}{\partial \theta}\right|_1 \Delta\theta + \cdots \tag{B.17}$$

With small excursion about the operating point, the higher derivatives in all parameters can be neglected as hence Equation (B.17) becomes:

$$F - F_1 \equiv \Delta F = \left.\frac{\partial F}{\partial P_S}\right|_1 \Delta P_S + \left.\frac{\partial F}{\partial \theta}\right|_1 \Delta\theta\,. \tag{B.18}$$

Defining

$$K_{P_S} = \frac{\partial F}{\partial P_S}\,, \tag{B.19}$$

$$K_\theta = \frac{\partial F}{\partial \theta}\,. \tag{B.20}$$

Substituting them to Equation (B.13) yields

$$T_{lp} = k_5 + k_6 P_s + K_{P_S} P_S + K_\theta\theta + k_8\dot{\theta} = K_5 + K_{6P_S} P_S + K_\theta\theta + K_8\dot{\theta}\,. \tag{B.21}$$

where $K_{6P_S} = K_6 + K_{P_S}$.

Substituting Equations B.12, B.21, B.14 and B.15 to Equation B.8, the T_{net} term is obtained as follows:

$$T_{net} = T_{yk} - T_{spr} - T_{yd} - T_{lp} = K_{10}P_c - K_3\theta - K_4 - K_9\dot{\theta} - K_5 - K_{6P_S}P_S - K_\theta\theta - K_8\dot{\theta}$$

$$= K_{10}P_C - K_{6P_S}P_S - K_{3\theta}\theta - K_{98}\dot{\theta} - K_{45}. \quad \text{(B.22)}$$

where $K_{3\theta} = K_3 + K_\theta$, $K_{98} = K_9 + K_8$, and $K_{45} = K_4 + K_5$.

Substituting Equation (B.22) to Equation (B.16):

$$K_{11}\ddot{\theta} = K_{10}P_C - K_{6P_S}P_S - K_{3\theta}\theta - K_{98}\dot{\theta} - K_{45}. \quad \text{(B.23)}$$

After the Laplace transformation, θ can be described as:

$$\theta(s) = \frac{K_{10}P_C(s) - K_{6P_S}P_S(s)}{K_{11}s^2 + K_{98}s + K_{3\theta}}. \quad \text{(B.24)}$$

Substituting Equations (B.10) and (B.11) into Equation (B.3), the output flow $Q_S(s)$ is obtained as:

$$Q_S(s) = K_1\theta(s) - k_2P(s). \quad \text{(B.25)}$$

Substituting Equation (B.24) to Equation (B.25), the system transfer function is found to be:

$$Q(s) = \frac{K_1K_{10}P_C(s) - (K_2K_{11}s^2 + K_2K_{98}s + K_2K_{3\theta} + K_1K_{6P_S})P_S(s)}{K_{11}s^2 + K_{98}s + k_{3\theta}}$$

$$= \frac{AP_C(s) - (Bs^2 + Cs + D)P_S(s)}{Fs^2 + Gs + H}. \quad \text{(B.26)}$$

where $A = K_1K_{10} = S_\omega K_\omega$,

$B = K_2K_{11} = R_l I_{yk}$,

$C = K_2K_{98} = K_2(K_9 + K_8) = R_l(R_{yd} + K_{pr4})$,

$$D = K_2 K_{3\theta} + K_1 K_{6P_S} = R_l (K_3 + K_\theta) + (S_\omega \cdot K_\omega)(K_6 + K_{P_S})$$

$$= R_l (K_{spr} + K\theta) + (S_\omega \cdot K_\omega)(K_{pr2} + K_{P_S}),$$

$$F = K_{11} = I_{yk},$$

$$G = K_{98} = K_9 + K_8 = R_{yd} + K_{pr4},$$

$$H = K_{3\theta} = K_3 + K_\theta = K_{spr} + K_\theta.$$